AN ABSTRACT OF THE THESIS OF

Wm. Forrest Hudson for the degree of Master of Science in Electrical and Computer Engineering presented on May 9, 1997.

Title: Experimental Verification of a Mismatch Shaping DAC.

Redacted for privacy

Abstract approved: _____

Richard Schreier

Delta-sigma data converters have gained popularity in both analog-to-digital and digital-to-analog converters (ADCs and DACs) due to their simplicity, high linearity and immunity to many analog circuit imperfections. These data converters include features such as oversampling, noise-shaping, and (historically) single-bit quantization. Single-bit converters are preferred for their inherent linearity. This is a feature which multibit converters cannot realize due to the unavoidable phenomenon of element mismatch. Because of this problem, multibit converters have been largely unexplored, and the market has seen few multibit commercial products.

Earlier work has shown that multibit DACs constructed with unit elements can be applied in an architecture which shapes the spectrum of the noise caused by element mismatch. The basis of this thesis is the experimental verification of such a DAC. A Xilinx 4005 FPGA is utilized to implement a 3^{rd}-order 4-bit delta-sigma modulator and the mismatch-shaping logic, while a custom IC consisting of 16 individually-controlled differential current sources implements the unit-element DAC. The final DAC achieves a Spurious Free Dynamic Range (SFDR) of 96 dB at a sampling rate of 62.5 kHz.

Experimental Verification of a Mismatch-Shaping DAC

by

Wm. Forrest Hudson

A Thesis Submitted
to
Oregon State University

In Partial Fulfillment of
the requirements for the
degree of

Master of Science

Presented May 9, 1997
Commencement June 1997

Master of Science thesis of Wm. Forrest Hudson presented on May 9, 1997

Approved: Redacted for privacy

Major Professor, representing Electrical and Computer Engineering

Redacted for privacy

Chair of Department of Electrical and ~~Computer~~ Engineering

Redacted for privacy

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for privacy

Wm. Forrest Hudson, Author

ACKNOWLEDGEMENTS

With the upmost respect and gratitude, I wish to thank my advisor Dr. Richard Schreier. My stay in Corvallis lasted four years, and I value my friendship with Richard as one of the highlights during this time. I must say that without his enthusiastic personality, attention to detail, and wonderful guidance this work would not have been possible.

I would also like to thank the members of my graduate committee for their time and effort: Professor Emeritus Don Amort (ECE), Professor Shih-Lien Lu (ECE), and Professor Janet Tate (PH).

Next, I must thank Matt Rupert for his help with making the circuit boards, Paul Stoffregen for helping me out with Intel Hex, Perl programming, and Xilinx downloading, and Jeff Stone for help with computer graphics.

I'd like to thank a few of my friends, lab partners, and colleges from school which I will always remember for reasons to numerous to list. We have in the order in which I met them, Zac Hosack, Bill Freer, Tom LePage, Ray Stein, Rick Lutz, and Mike Setera.

Thanks to the Delta-Sigma gang: Bo Zhang, Haiqing Lin, Tao Shui, Min Shen, and special thanks to Prof. Luis Hernandez.

I would also like to thank my lovely wife Lisa for her caring support.

And finally, without the support of my parents, Anne & Bill Hudson, I wouldn't be where I am today. Here's to you, Mom & Dad, you're the greatest!

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

Experimental Verification of a Mismatch-Shaping DAC

Chapter 1. Introduction

This thesis builds on earlier theoretical work which has shown that multibit delta-sigma data converters are feasible despite the loss of the inherent linearity property of single-bit DACs. The goal is to demonstrate noise-shaping of the nonlinearity errors in a multibit DAC with a prototype system.

1.1 Motivation

In general, real-world signals are of an analog nature. Thus analog signal processing would appear to be the natural choice. Indeed, analog signal processing circuits dominated electronics in the early days. However, digital-signal-processing is nowadays generally preferred over analog signal processing for a large number of reasons. In comparison with analog circuits, digital signal processing circuits are cheaper, more reliable, and more easily modified. The rapid developments in very-large-scale integration (VLSI) of electronic circuits have made it possible to construct sophisticated and inexpensive digital signal processing circuits. This technology has increased the demand for analog-to-digital and digital-to-analog converters which serve as the interface between the analog and digital worlds.

Delta-sigma data converters have gained popularity as both analog-to-digital and digital-to-analog converters due to their simplicity, high linearity and tolerance of analog circuit imperfections. These data converters include features such as oversampling, noise-shaping, and (historically) single-bit quantization. Although most delta-sigma modulators adopt a one-bit quantizer because a one-bit DAC is inherently linear, multibit quantization

enhances modulator performance. Multibit modulators can accommodate nearly full scale inputs, can employ noise transfer functions with large out-of-band gains, and can exploit higher-order loop filters. Unfortunately, the DAC linearity required for these enhancements to be realized is usually well beyond that which is practical. Nonetheless, the theoretical performance afforded by multibit quantizers has driven the search for ways to exploit multibit quantization.

1.2 Outline of the Thesis

Chapter 2 provides the reader with the necessary background information on delta-sigma modulation. Chapter 3 characterizes a custom IC called the *ueDAC* and determines values for several performance-critical parameters. Chapter 4 details the implementation and testing of a multibit modulator. Experimental results demonstrating shaping, comparisons with simulations, and explanations of discrepancies are presented. Chapter 5 concludes the thesis and gives directions for future work.

Chapter 2. Delta-Sigma Background

This chapter presents the material necessary to understand both delta-sigma modulators in general and the particular modulator used in this thesis. Starting with a brief review of digital-to-analog conversion, the discussion progresses to first-order, single-bit modulators and then to higher-order, multibit modulators.

2.1 Digital-to-Analog Conversion

Digital-to-analog converters (DACs) [1] serve as the interface between the discrete-time digital world and the continuous-time analog world. In short, DACs convert a digital word into an analog quantity, such as a voltage or current, and analog-to-digital converters (ADCs) do the reverse.

In a conventional Nyquist-rate ADC or DAC, higher resolution is achieved by using smaller step sizes. However, small steps require the use of precisely-matched analog components. For example, suppose that we wish to design a 16-bit DAC. To do so with an R-2R ladder would require resistor matching on the order of 1 part in 2^{16} (about 16ppm), which is an order of magnitude beyond the capability of current technology.

As a result, the practical limit with current (untrimmed) circuit techniques is about 14 bits of resolution. Trimmed circuits can achieve 16 or more bits of resolution, but are expensive. In a Nyquist-rate ADC or DAC, precision analog circuits, such as high-gain op-amps, linear integrators, etc., have no opportunity to exert their power because a complete conversion must be performed in every clock period. Delta-sigma modulation increases the sampling rate (above the Nyquist rate) to provide the freedom that allows the features of precision analog circuits to be exploited.

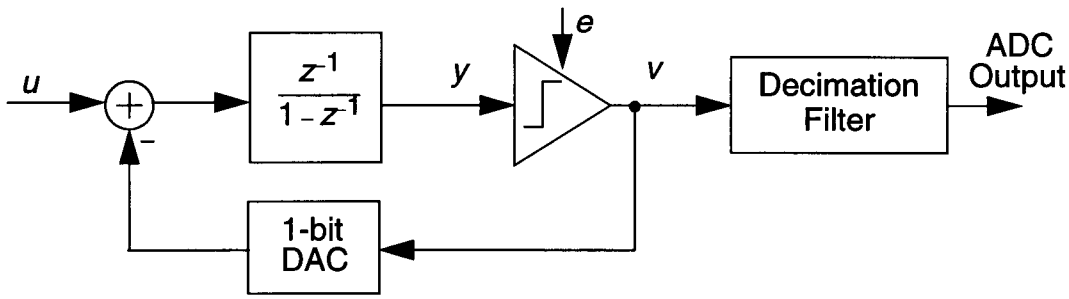


Figure 2.1: MOD1: A first-order delta-sigma ADC.

Oversampling [2] is simply the process of sampling faster than the Nyquist criterion requires. If the signal occupies the band from DC to f_B, the Nyquist criterion requires $f_s > 2f_B$, where f_s is the sampling rate; the oversampling ratio OSR is then defined as $OSR = f_s/(2f_B)$. One advantage of oversampling is that it eases the anti-alias filter design since a wide transition band is created by the increased separation between the signal band and its first alias. For an ADC with broadband quantization noise, oversampling also reduces the amount of in-band quantization noise. This allows the conversion to be more accurate than the resolution of the quantizer. Specifically, an increase in resolution of 0.5 bits results from each octave increase in the oversampling ratio. As the next section will show, delta-sigma modulation improves significantly on this trade-off.

2.2 MOD1: The First-Order Delta-Sigma Modulator

A first-order delta-sigma ($\Delta\Sigma$) converter (MOD1) [2], consisting of an analog integrator, a single-bit quantizer (comparator), a single-bit DAC and a digital decimation filter, is shown in Figure 2.1. One may view the "delta" and "sigma" as referring to the analog operations in the system loop: subtraction of the fed-back output signal from the input signal and accumulation (integration) of the differences. In order for the output of the integrator to be bounded, the DC component of the feedback must be exactly the same as that of the input signal. If this is true, the first-order converter has ideally unlimited

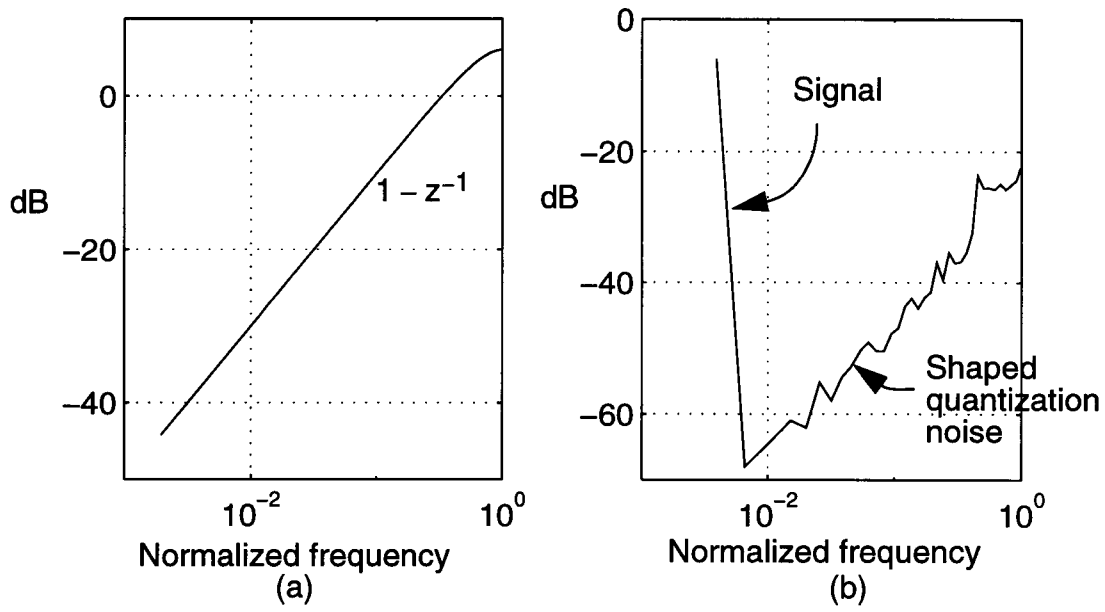


Figure 2.2: (a) The frequency response of $1 - z^{-1}$, and (b) the simulated spectrum of first-order delta-sigma modulator with sine-wave input shows the shaping of quantization noise.

resolution, at least for DC signals. The digital lowpass decimation filter removes the out-of-band noise and produces a high-resolution digital representation of the input.

The output of MOD1 in the z domain is

$$V(z) = z^{-1}U(z) + (1 - z^{-1})E(z) , \qquad (2.1)$$

where V is a discrete-time binary-valued signal, U is a discrete-time continuous-amplitude signal and E is the quantization error[3]. According to Eq. (2.1), the quantization error is frequency-shaped by the function $H(z) = 1 - z^{-1}$. This *noise transfer function* (NTF) has a zero at DC and thus suppresses the quantization noise in the vicinity of DC. The spectrum of the output of MOD1 in Figure 2.2 clearly shows this effect.

Assuming that E is white with power σ_e^2, the in-band noise power for MOD1 is

$$N_0^2 = \frac{\sigma_e^2}{\pi}\int_0^{\frac{\pi}{OSR}} |H(e^{j\omega})|^2 d\omega \approx \frac{\sigma_e^2}{\pi}\int_0^{\frac{\pi}{OSR}} \omega^2 d\omega = \frac{\sigma_e^2 \pi^2}{3(OSR)^3}, \qquad (2.2)$$

where $\sigma_e^2 = \frac{\Delta^2}{12} = \frac{1}{3}$, if E is uniformly distributed in $[-1, 1]$.

The above equation predicts that an octave (factor of two) increase in *OSR* will increase the SNR by 9 dB (which is 1.5 bits by the 6 dB = 1 bit rule). With an oversampling ratio of 100, the rms noise level in the band of interest should be on the order of 10^{-3}, or -60 dB. In principle, the in-band noise can be made as small as desired, simply by making *OSR* large enough. Thus, MOD1 has potentially unlimited accuracy, independent of component mismatch and many other non-idealities. In general, the resolution of a delta-sigma converter is improved by clocking faster (which is easy) and not by making larger, more sensitive analog circuitry (which is hard). In practice, the achievable linearity is limited by finite op-amp DC gain and the linearity of the analog components.

An important property of single-bit modulators is what is often referred to as "inherent linearity" [4]. This property comes from the fact that the input-output transfer curve of any static two-level DAC can be modeled exactly by a straight line joining the two points on the curve. A binary DAC is therefore ideal and cannot introduce errors other than simple offset and gain errors. These errors do not introduce distortion and the conversion is "linear."

The primary disadvantage of MOD1 is that a high oversampling ratio is needed to achieve high resolution. For example, if we want 16-bit resolution, the oversampling ratio must be about 1500. Except for very low-frequency applications, a high oversampling ratio leads to a high sampling frequency and thus difficulties in implementation. The

oversampling ratio required to achieve a given resolution can be made smaller if higher-order delta-sigma modulators are used, but modulator instability then becomes a problem.

Multibit quantization is in theory the most attractive way to increase a modulator's resolution since it increases resolution while simultaneously reducing the modulator's susceptibility to instability. Furthermore, multibit modulators can accommodate nearly full scale inputs, can employ NTFs with large out of band gains, and can exploit higher order loop filters. However, a highly-linear multibit DAC is needed to prevent distortion in a multibit system. Unfortunately, the DAC linearity required for these enhancements to be realized is usually well beyond that which is practical.

2.3 Theory of Operation

Multibit quantization is generally avoided in delta-sigma modulators because nonlinearity in the multibit DAC is translated directly into nonlinearity for the overall converter, (the DAC is directly in the signal path). However, recently reported simulation results indicated that element mismatch errors in a multibit DAC constructed from unit elements can be noise-shaped [5], [6]. A related work illuminated the operating principle and showed that first-order mismatch-shaping was the result [7].

Figure 2.3 shows a multibit delta-sigma DAC with the proposed mismatch-shaping DAC. The upper portion of the diagram[8] depicts an ordinary delta-sigma modulator realized with the error-feedback structure, while the lower portion depicts the key part of the mismatch-shaped multibit DAC: the element selection logic. The two blocks are drawn in a manner which emphasizes their similarity. A key difference between the two blocks is that many signals in the element selection logic are vectors; these are denoted with bold text and heavy lines. The output of the selection logic controls the unit-element DAC and the

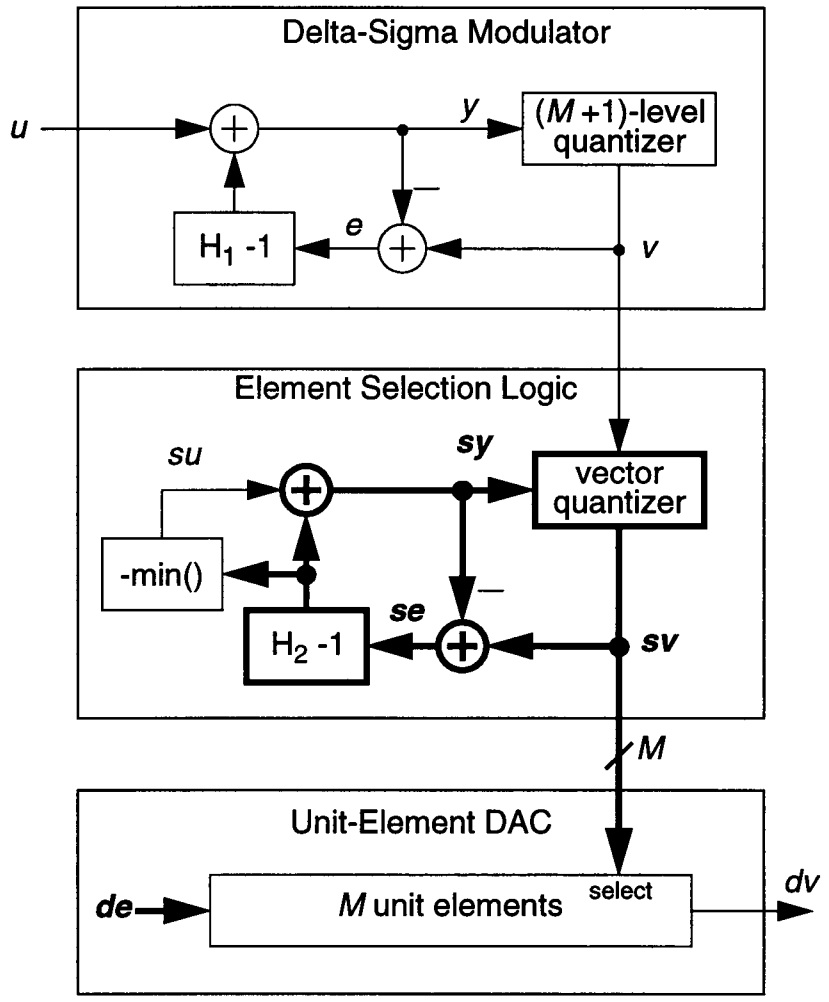


Figure 2.3: The system diagram of a multibit delta-sigma DAC with a digital delta-sigma modulator, an element-selection logic and a unit-element DAC.

selected elements are summed to form the output of DAC. Figure 2.4 is a conceptual diagram of an M+1 level DAC employing M unit elements.

The modulator block accepts a finely-quantized signal, u, and produces a coarsely-quantized signal, v. Denoting the Z-transform of quantizer error by $E(z)$, the output of the modulator is:

$$V(z) = U(z) + H_1(z)E(z), \tag{2.3}$$

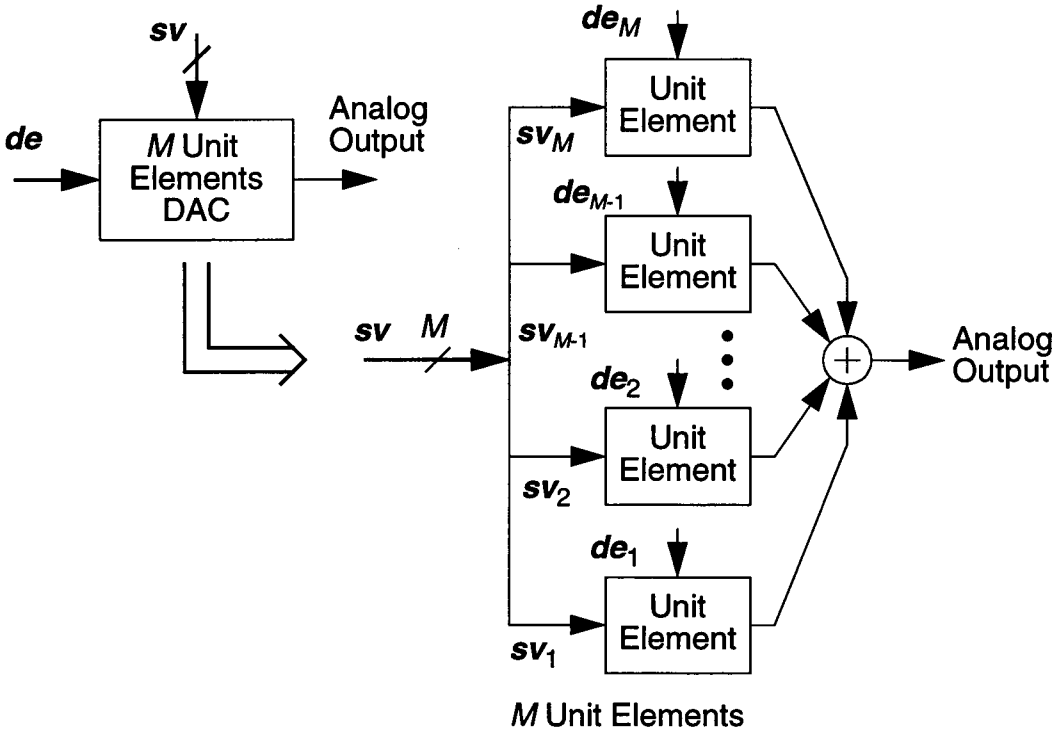


Figure 2.4: A conceptual diagram of an M+1 level DAC employing unit elements.

Thus, the output of the modulator is equal to its input plus an error term which, by suitable choice of H_1, the NTF of modulator, can be designed to have a small magnitude in a selected frequency range. For the purpose of this discussion, assume that v is quantized to one of the M+1 integers in [*0, M*]. Using multibit quantization ($M \geq 2^N$), a high-order NTF $H_1(z) = (1 - z^{-1})^N$, can be realized without regard to the restriction imposed by the Lee criterion [9]; the stability of the modulator is guaranteed.

At each time step n, the element selection logic determines which $v(n)$ of the M unit elements will be used to form the analog output value. The output of the selection logic is $\boldsymbol{sv}(n)$, a $1 \times M$ vector containing $v(n)$ ones and M-$v(n)$ zeros. Each unit element in the DAC

is controlled by a specific component of $\boldsymbol{sv}$, so the output of the DAC will be an analog version of $v(n)$ plus an error term due to element mismatch. It is the function of the selection logic to ensure that the error term has a mismatch-shaped spectrum.

The selection vector, $\boldsymbol{sv}$, is computed in a manner analogous to that which produces v, namely each component of $\boldsymbol{sv}$ is the output of a modified delta-sigma modulator realized with the error feedback structure. The main modification is in the vector quantizer, but this modification does not affect the basic operation of the element selection logic. A discussion of the vector quantizer and the su input will be delayed until later.

Given that $\boldsymbol{sv}$ is computed in a manner analogous to that which produces v, the output of the selection logic may be written immediately as

$$\boldsymbol{SV}(z) = SU(z)[1 \ldots 1] + H_2(z)\boldsymbol{SE}(z) \tag{2.4}$$

Now, let $\boldsymbol{de}$ be a (fixed) $1 \times M$ vector containing the difference between the value of each unit element in the DAC and the mean of all the elements. An immediate consequence of this definition is that the sum of the components of $\boldsymbol{de}$ is precisely zero:

$$[1 \ \ldots \ 1] \bullet \boldsymbol{de} = 0, \tag{2.5}$$

where $\boldsymbol{a} \bullet \boldsymbol{b}$ represents the scalar (dot) product of two vectors.

Since the error between the actual output of the DAC and its ideal output is $\boldsymbol{sv} \bullet \boldsymbol{de}$, the DAC error is

$$SV(z) \bullet de = \left[SU(z)[1 \ \ldots \ 1] + H_2(z)\boldsymbol{SE}(z)\right] \bullet \boldsymbol{de} = SU(z) \cdot 0 + H_2(z)(\boldsymbol{SE}(z) \bullet \boldsymbol{de})$$
$$= H_2(z)(\boldsymbol{SE}(z) \bullet \boldsymbol{de}). \tag{2.6}$$

The previous equation shows that static DAC errors are shaped by the transfer function H_2, provided, of course, that the ***se*** signal is bounded. If ***se*** is bounded, the mismatch-shaping implied by Eq. (2.6) is a result that is independent of the *su* input signal, the operation of the vector quantizer, and most importantly, the errors in the unit elements.

Based on the element usage requirement *v(n)* and on the contents of the vector ***sy***(*n*), the vector quantizer sets certain elements of ***sv****(n)* to one. The error of this quantization operation, ***se****(n)*, is fed back through an array of H_2-1 filters and added to the scalar-valued selection logic input, *su(n)* to form subsequent samples of ***sy***. Choosing *su*(*n*) to be equal to the negative minimum value of the filter outputs makes all components of ***sy****(n)* non-negative, with the smallest component equal to zero. Note that the addition of a constant to all elements in the ***sy*** vector does not disturb the noise-shaping property of the selection logic; its purpose is simply to reduce the magnitude of the signals which need to be stored and processed.

The vector quantizer decides which elements of ***sv*** should be set to one. Many vector quantizer algorithms are possible, but the one which was first proposed sets those elements of ***sv****(n)* to one which correspond to the *v(n)* largest components of ***sy****(n)*.

From Eq. (2.6), we see that the analog output of the DAC is described by

$$DV(z) = U(z) + H_1(z)E(z) + H_2(z)(\boldsymbol{SE}(z) \bullet \boldsymbol{de}) \qquad (2.7)$$

This equation shows that the DAC output is equal to the modulator input plus two error terms. The first error term is the quantization noise of the $(M+1)$-level quantizer, shaped by the noise transfer function H_1 of the modulator. The second error term is the element mismatch error, multiplied by the selection error and shaped by the mismatch-shaping function H_2.

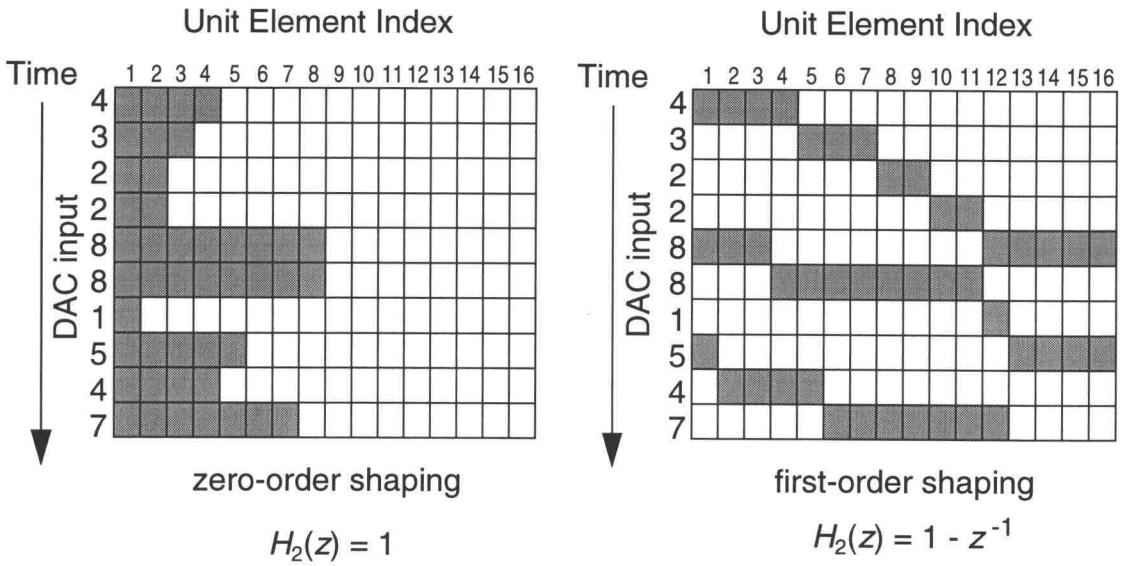


Figure 2.5: The unit element selection patterns in a zero-order and first-order mismatch-shaped 4-bit DAC. Each box represents a unit element. The numbers on the left are the number of elements that are to be enabled at each time step and the shaded boxes indicate which elements are enabled.

Like regular delta-sigma modulators, the selection logic loop around the vector quantizer cannot be delay-free, so $H_2 - 1$ must be strictly causal [9][10] (first impulse-response coefficient zero). This consideration results in the familiar realizability constraint,

$$H_2(\infty) = 1. \tag{2.8}$$

Furthermore, like regular delta-sigma modulators, the element selection logic is subject to instability (characterized by *se* becoming unbounded). The *su* input sequence, the value of H_2 and the quantizer algorithm all affect the stability of the element selection logic. As a result, proving the stability of the selection logic is a much more difficult problem than proving the stability of a simple binary delta-sigma modulator. Despite many years of effort the latter problem is still unsolved, so the stability of general ESL is likely to remain unknown for many years to come.

2.4 First-Order Mismatch-Shaping

Figure 2.5 shows the unit-element selection patterns of a zero-order ($H_2(z) = 1$) and first-order ($H_2(z) = (1 - z^{-1})$) mismatch-shaped 4-bit DAC. These two ESL schemes will be used in Chapter 4. Zero-order shaping is simply the conventional DAC selection: select the first $v(n)$ elements in the array. No state information is needed to implement this DAC. The first-order shaping pattern can be considered to be integration followed by differentiation with a *modulo* operation, so that the elements are chosen in a circular fashion, starting from the element adjacent to that which was most recently used. This scheme is functionally identical to that of [5], [6]. Consequently an implementation of first-order shaping of the element mismatch noise is trivial, requiring only one register of length $\log_2 M$ bits and some combinational logic. It is also comforting that the stability of 1^{st}-order ESL is guaranteed as long as $0 \leq v(n) \leq M$.

2.5 Conclusion

Delta-sigma modulation is becoming the preferred method for high-accuracy analog-to-digital and digital-to-analog conversion because of the high degree of linearity that can be achieved using analog circuits of moderate precision. Multibit quantization has been shunned because it destroys the inherent linearity property of single-bit modulators. Schemes for shaping the mismatch errors in multibit DACs have been proposed and demonstrated with simulations, but experimental verification has usually been left as an exercise for the reader. This thesis carries out this exercise.

Chapter 3. Chip Characterization

This chapter is an application note on José da Silva's chip, the *ueDAC* [11]. It provides the raw data for the analysis of Chapter 4 as well as the information needed to use the *ueDAC* in alternative designs.

3.1 General Description

The *ueDAC* is a current-mode DAC fabricated in the ORBIT 1.2 μm N-well process. The DAC consists of 16 (ideally) equal-valued current sources. As shown in Figure 3.1, all current sources can be switched to either of two output nodes. The current sources are implemented using a regulated-cascode configuration to ensure high output impedance and differential drivers yield fast switching. Figure 3.2 shows a pin-out of the packaged chip and Table 3.1 provides a brief description of each pin.

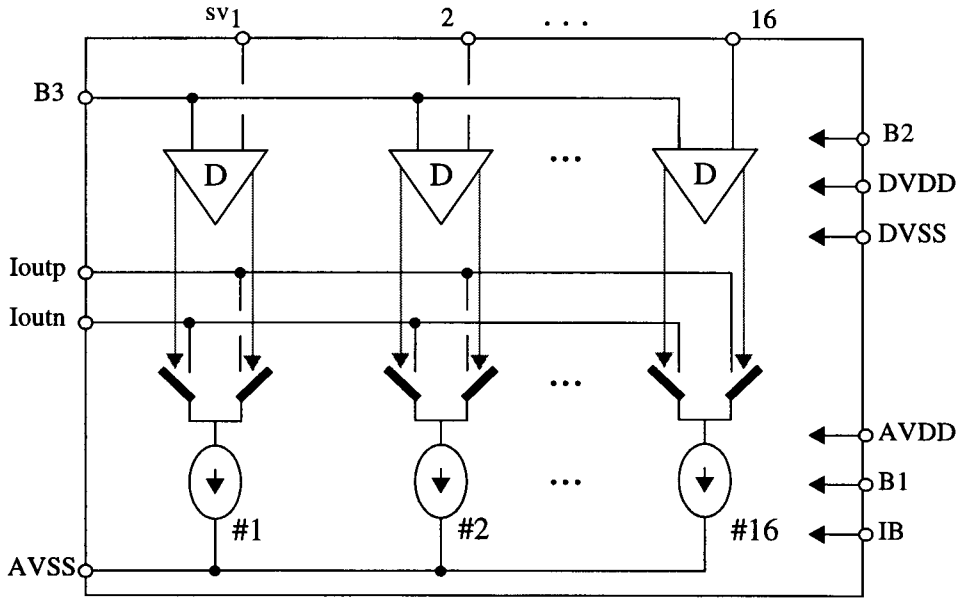


Figure 3.1: The *ueDAC* functional block diagram.

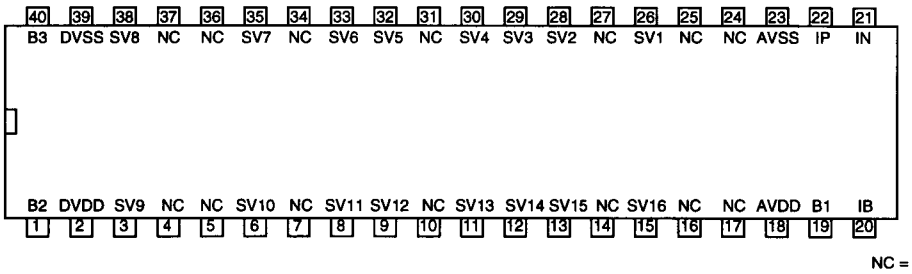


Figure 3.2: Connection diagram: Top view.

Table 3.1: Pin Description

Pin	Signal	Description	Value
18	AVDD	Power supply for analog circuitry	5V
23	AVSS	Analog ground	GND
2	DVDD	Power supply for digital circuitry	5V
39	DVSS	Digital ground	GND
1	B2	Bias for driver current source	650μA; 2.5V[(1)]
40	B3	Threshold for digital input	2.5V (nominal)
	sv	16 bit digital input	Digital
19	B1	Bias for the regulated cascode loop amplifiers	180μA; 2.5V[(1)]
20	IB	Unit element current source value	100μA; 2.08V[(1)]
21	Ioutn	Negative output current	0-16 times IB
22	Ioutp	Positive output current	0-16 times IB

[(1)] Either a voltage or a current can be applied. If one is used, the other should be verified.

3.2 Circuit Description

The *ueDAC* chip consists of 16 differential regulated-cascode current sources which we will call unit elements. Figure 3.3 shows the schematic of one unit element together with its clock driver circuit. A unit element will drive one of the two output nodes according to its digital input bit *sv*. The current sources are nominally equal-valued, so that

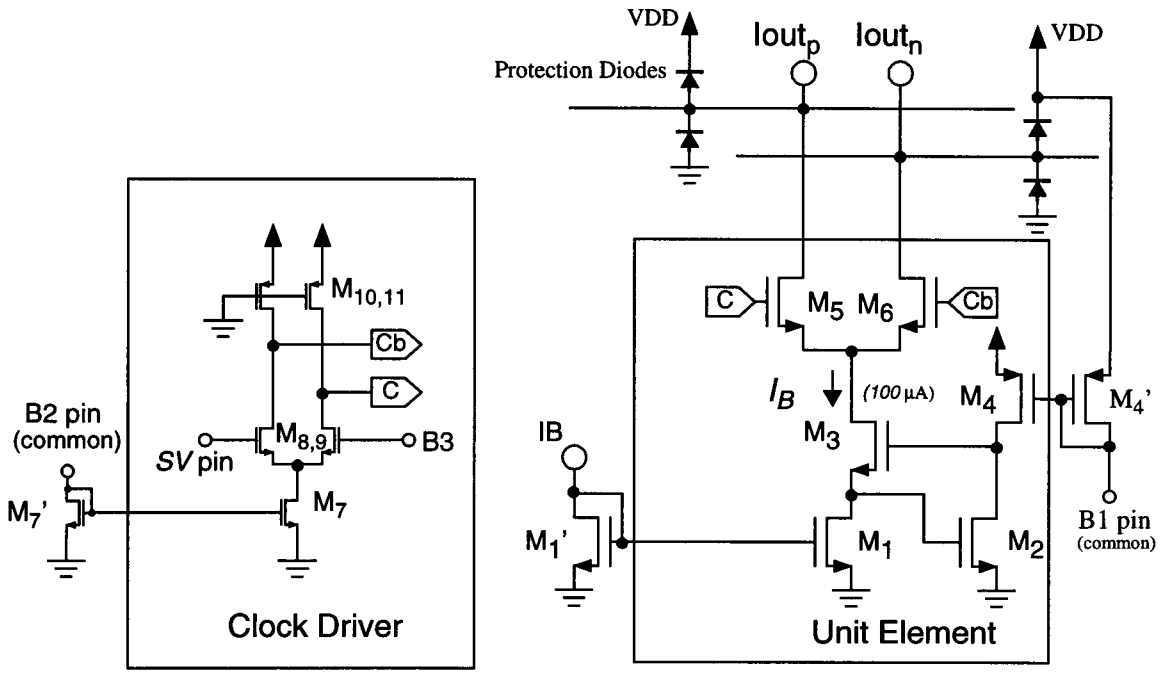


Figure 3.3: Schematic of the unit element & clock driver circuit.

in contrast to a commercial binary weighted DAC, the output is not weighted. Ideally, the output currents are related to the inputs according to the following equations:

$$Iout_p = I_B \text{ x } \sum_{i=1}^{16} sv_i \tag{3.1}$$

$$Iout_n = I_B \text{ x } (16 - \sum_{i=1}^{16} sv_i). \tag{3.2}$$

Table 3.2: Transistor Dimensions

M_1	30/10
$M_{2,3}$	50/3
M_4	4.2/1.2
$M_{5,6}$	10/1.2
M_7	20/2
$M_{8,9}$	50/1.2
$M_{10,11}$	7/1.2

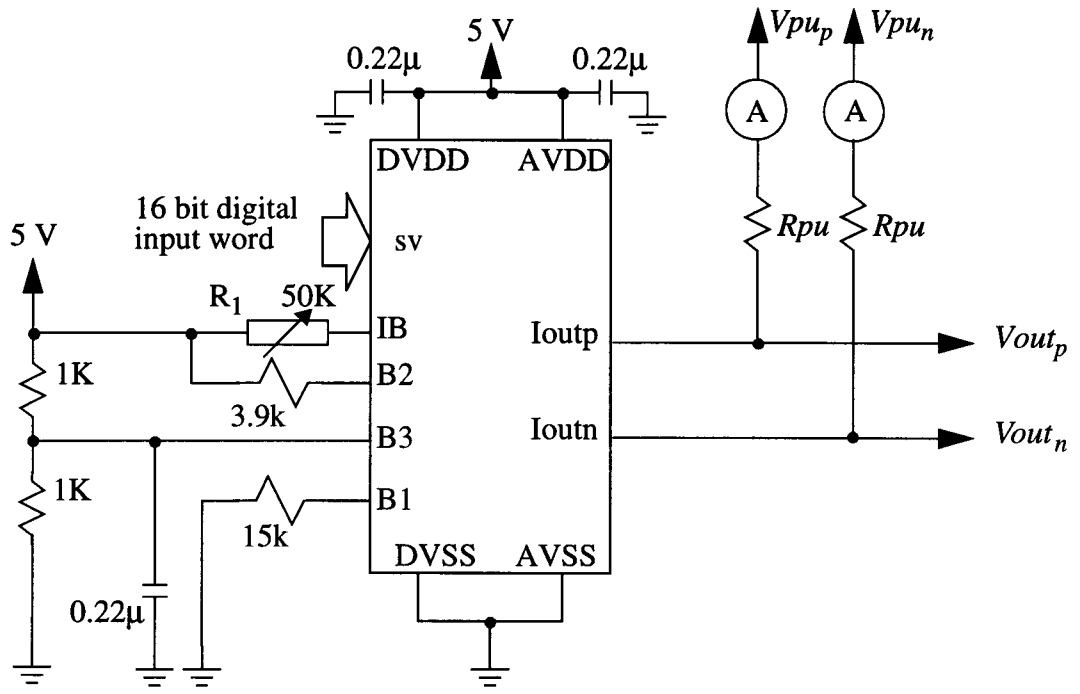


Figure 3.4: Test setup for static chip characterization.

Eq. (3.1) and (3.2) show that in theory the output is independent of *which sv* bits are turned on, as long as the total number of *sv* bits turned on is fixed. Unfortunately, the unavoidable mismatch between the current sources will cause the output to deviate from this ideal behavior.

3.3 Application Notes

Figure 3.4 shows the circuit used for static testing. The value of the adjustable resistor R_1 defines the step between the output levels. In this chip a value of about 29 kΩ yields a 100 µA step. The B3 voltage defines a threshold for the input drivers. In most cases 2.5 V is a logical and convenient choice. Note that B1-B3 are shared among all elements and that by virtue of the M7' and M4' transistors, B1 and B2 can be programmed using either a voltage or current.

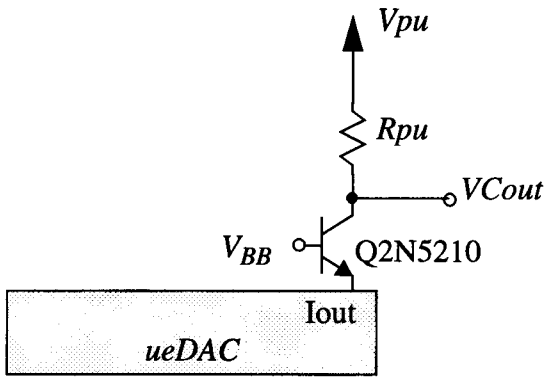


Figure 3.5: Using a cascode configuration for high speed.

When using pull-up resistors to convert the *ueDAC's* current output to a voltage, care must be taken in selecting the resistor values. The output current sources have a parasitic capacitance, so small values of *Rpu* will enable faster operation by reducing the voltage swing at the outputs. Unfortunately this also means that the output signal is small.

The next consideration is to make sure the unit elements always work in the proper region. At IB = 100 μA, experiments show that keeping the unit element transistors saturated requires that the voltage on Ioutp or Ioutn be at least *Vmin* = 1.5 V.

In order to keep the transistors working in the proper region, *Vpu*, IB, and *Rpu* should be set such that the following condition holds:

$$Vpu - 16 \times IB \times Rpu > Vmin . \qquad (3.3)$$

For example, if *Vmin* = 1.5 Volts, *Vpu* = 5 V and *IB* = 100 μA, then *Rpu* < 2.2 kΩ.

The outputs can also be connected to bipolar transistors in a cascode configuration as shown in Figure 3.5. Theoretically, a higher speed of operation can be achieved with this circuit, since it eliminates the parasitic capacitance on the output node due to the IC pad and the internal current sources. This configuration also allows *Vpu* to be greater than VDD.

In order to assure proper operation for this configuration, *VCout* cannot drop below (V_{BB} - 0.2 V). It is also necessary that $V_{BB} > 0.7 + Vmin = 2.2$ V for IB = 100 μA (assuming a 0.7 V_{BE} on the transistor) in order to ensure that the unit elements continue to act as fixed current sources. Maintaining the following condition will ensure that the transistors will all stay working in the proper region.

$$Rpu < \frac{Vpu - V_{BB} - 0.2}{16 \times IB} \tag{3.4}$$

As an example, if $Vpu = 5$ V, $V_{BB} = 2.5$ V and IB = 100 μA, then $Rpu < 1.4$ kΩ.

3.4 Static Testing

Static testing was accomplished by building the circuit shown in Figure 3.3 on a breadboard. The data in Table 3.3 were obtained by individually selecting the *sv* inputs with dip-switches and measuring the current with a Fluke 79 multimeter.

Table 3.3: Static Testing of the current sources

sv	Ioutp (uA)	Ioutn (uA)
1	100.4	100.6
2	101.5	101.7
3	101.6	101.8
4	102.0	102.2
5	102.1	102.3
6	102.5	102.7
7	102.8	103.0
8	102.4	102.6
9	102.4	102.6
10	102.7	102.9
11	102.1	102.3
12	102.0	102.2
13	101.8	102.0
14	101.6	101.8
15	101.7	101.9
16	100.5	100.7

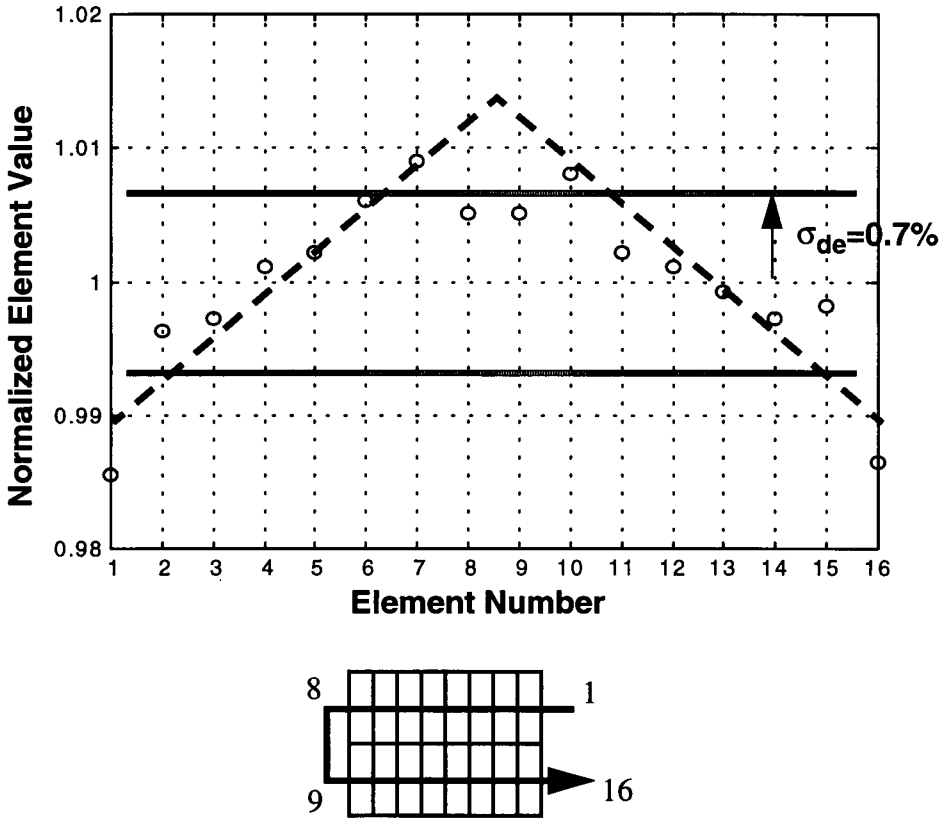


Figure 3.6: Matching of the sv current sources. The trend reflects the layout as shown above.

For a given element, Ioutp should equal Ioutn. The 0.2 μA difference between the two columns of data is probably due to a fixed leakage path on the board.

Analyzing the test results further shows that the unit elements have a mean output of 101.9 μA with a standard deviation of 0.68 μA. When these results are plotted, several interesting things can be observed (see Fig. 3.6). It would seem that aside from samples 1, 8, 9, and 16, the data indicates one of the most common albeit undesirable effects in CMOS, which is a gradient in the oxide layer. Both the gradient and the anomalous data points can best be explained by the layout. Note that as we proceed from right to left on the layout, the corresponding normalized output current increases. Likewise if we proceed from left to

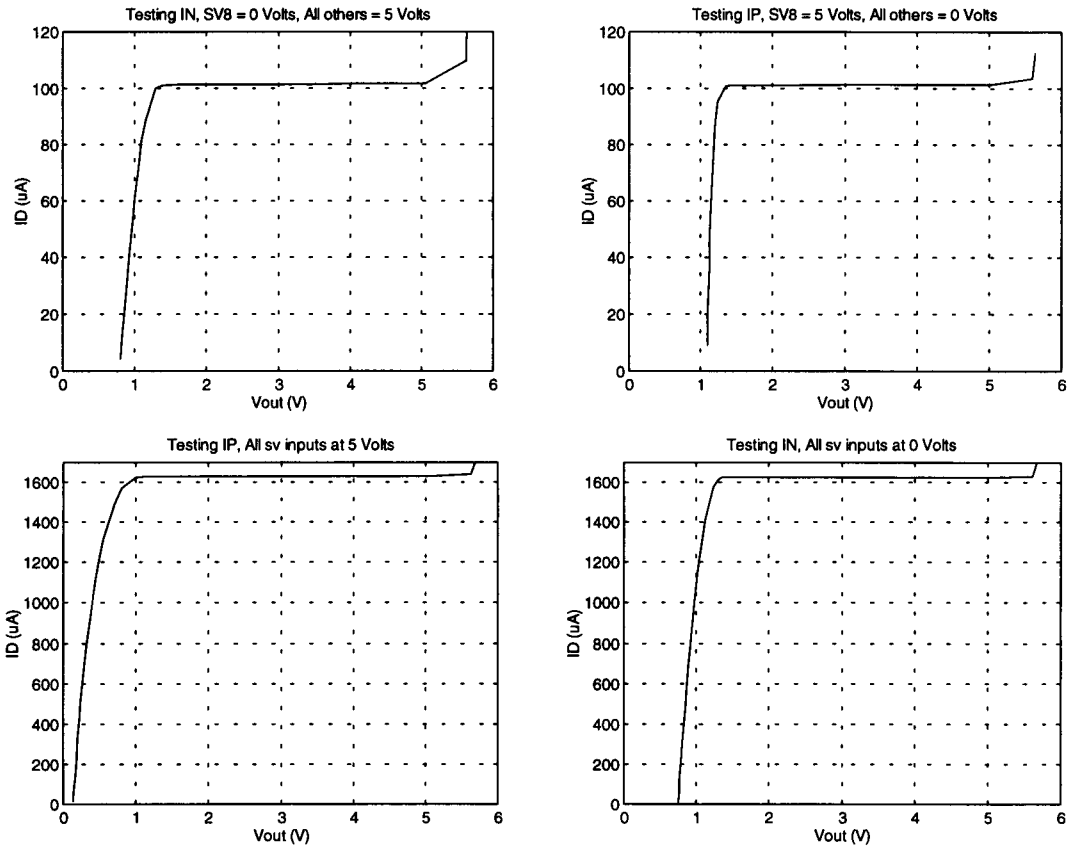


Figure 3.7: Plots showing $Iout_n$ and $Iout_p$ vs. *Vout* for different *sv* settings.

right, i.e. elements 9 to 16, the output current decreases. Thus, the simple arrangement of the current sources provides an explanation for the experimental results. The next interesting observation is that the incongruous data points all lie on the edges. This would indicate that edge effects also play a significant role in the observed transistor matching.

3.5 Output I-V Characteristics

The output characteristics of the current sources were determined using the circuit shown in Figure 3.4 with Rpu = 560 Ω. Testing the current sources was performed by setting $Vpu_{[n,p]}$ to the 5 V VDD supply and slowly sweeping $Vpu_{[p,n]}$ while observing the drain current and the corresponding output voltage $Vout_{[p,n]}$. To ensure accurate results with these sensitive measurements, after the onset of the saturation region the drain voltage

was recorded at precisely the point when an increase (as limited by the resolution of 0.1 uA from the Fluke) was noted in the current. Since the internal current sources are connected to the same node, V-I characterization tests were performed on both outputs, first with a single sv activated, and then second with all sv's activated. A low logic signal activates *sv* when testing Ioutn, and a high logic signal activates *sv* when testing Ioutp. Complete tabular results are contained in Appendix A.

On a side note, both outputs need to be connected to a pull-up voltage (through *Rpu*) because the chip does not work properly if one of the current sources is left floating. As was mentioned, the output which was not being tested was left connected to 5 Volts through its pull up resistor.

The important information learned from these results is that *Vmin*, the minimum output voltage needed to bring the unit elements into saturation when IB = 100 μA is about 1.5 Volts. Also note that *Vout* must be kept below 5.5 Volts (and above -0.5 Volts) to avoid forward biasing the internal protection diodes. This effect appears in the plots as a dramatic increase in Id for *Vout* > 5.5 V.

The slope of the line while the transistor is saturated yields the output resistance of a unit element. Equations 3.5 and 3.6 below use the numerical data taken (taken from the Appendix) when a single *sv* element is activated, and equations 3.6 and 3.7 use the data with all *sv* elements activated.

$$Rout_N = \frac{\Delta V}{\Delta I} = \frac{(5.06 - 1.642)\,V}{(101.7 - 101.3)uA} \cong 8.6M\Omega \pm 25\% \tag{3.5}$$

$$Rout_P = \frac{\Delta V}{\Delta I} = \frac{(5.02 - 1.615)V}{(101.4 - 101.1)uA} \cong 11M\Omega \pm 33\% \tag{3.6}$$

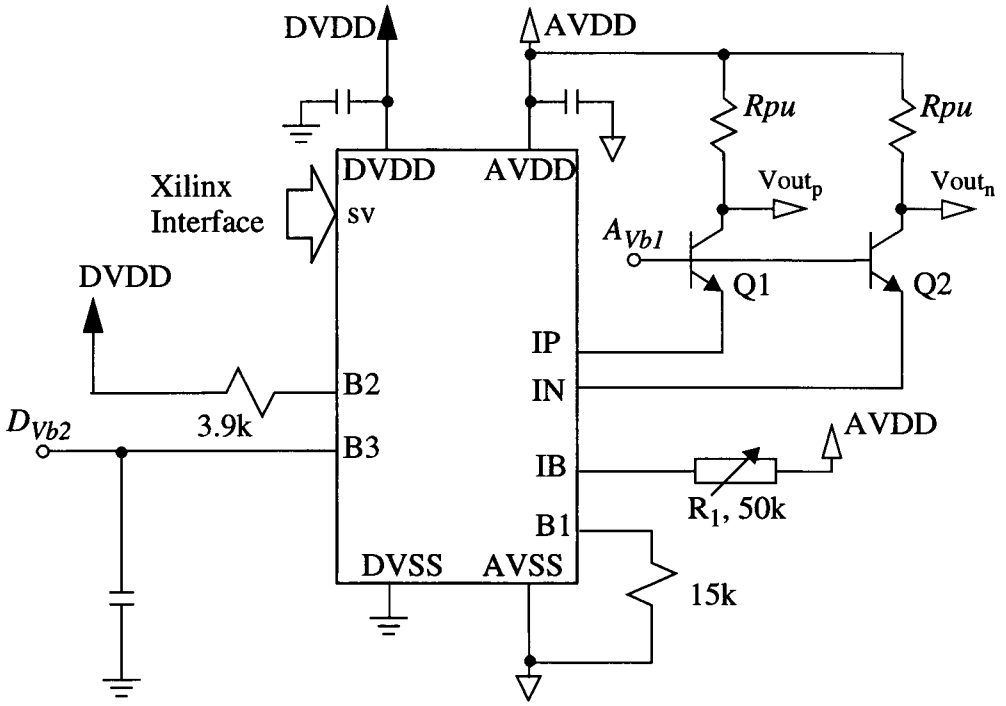


Figure 3.8: Simplified AC test circuit, Q1 and Q2 should be low-noise small-signal parts such as the Q2N5210.

$$Rout_N = \frac{\Delta V}{\Delta I} = \frac{(5.00 - 2.545)V}{(0.2)uA} \cong 13M\Omega \pm 50\% \tag{3.7}$$

$$Rout_P = \frac{\Delta V}{\Delta I} = \frac{(5.06 - 1.642)V}{(0.1)uA} \cong 10M\Omega \pm 100\% \tag{3.8}$$

From (3.5) and (3.6), we see that the output resistance of a current source is near the limit of measurability at about 10 MΩ. Since the output resistance of all the sources in parallel is also approximately 10 MΩ, this indicates that R_{out} is not due to the current sources themselves; it is in fact just the 10 MΩ input resistance of the voltmeter!

3.6 AC Test Results

An AC test circuit is more complicated than a DC test circuit. In this case, using a breadboard is unacceptable. Achieving good results requires following a few well-

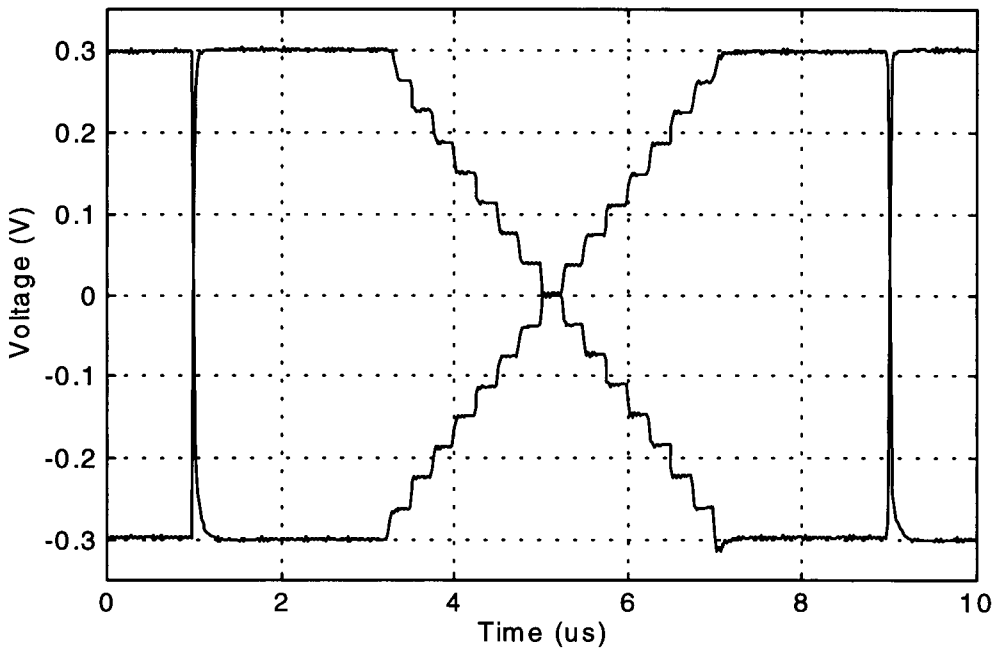


Figure 3.9: Discrete levels of Ioutn and Ioutp; AC coupling.

established guidelines regarding the circuit construction. First and foremost, low-noise designs require fabricating a PC board. Other common techniques include separating analog and digital ground planes, separating the analog VDD from the digital VDD, using proper decoupling capacitors, and keeping sensitive analog traces away from noisy digital ones. Figure 3.8 shows the simplified AC test circuit. Ch. 4 goes into greater details about the actual circuit board constructed for the AC testing. AC tests were performed by configuring a Xilinx 4005 to drive the *sv* inputs with selected waveforms.

The first test was to verify that the output achieves 16 discrete levels. To do this, the Xilinx was configured to drive the *sv* lines with a 16-level decoder, and the outputs were sampled with a Tek TDS 420 Scope. Figure 3.9 shows that all of the outputs are working as expected.

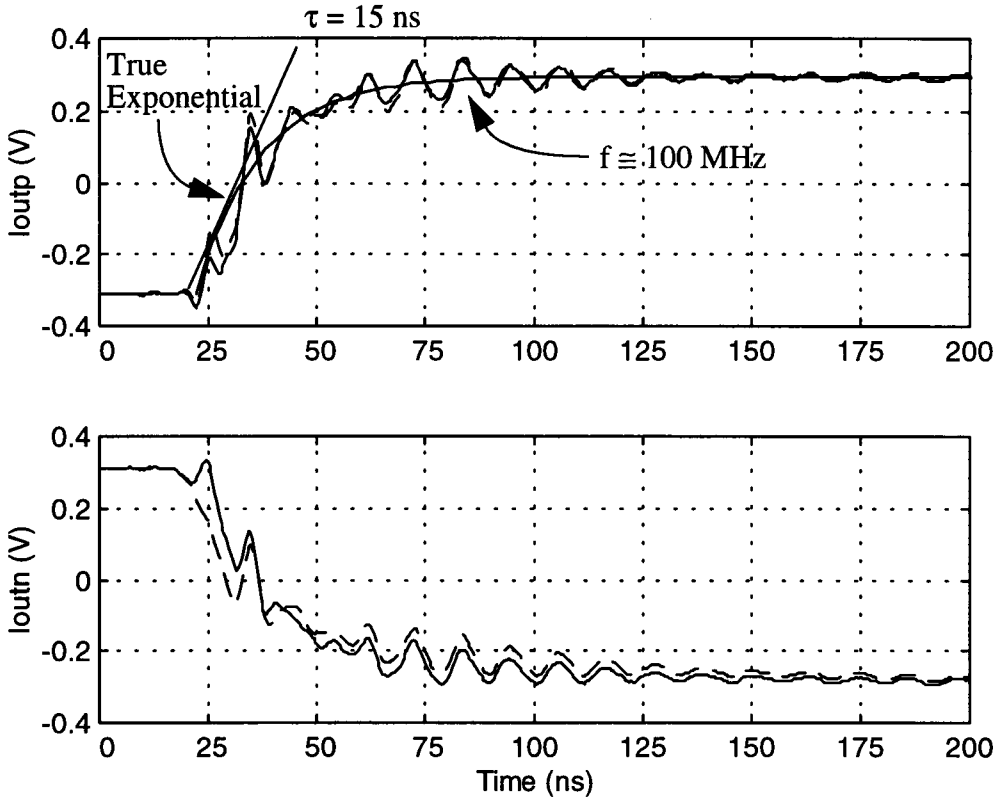


Figure 3.10: Full-scale step response of the *ueDAC* with *Rpu* = 371 Ω. Dashed lines are the results without the cascode transistor.

3.7 Large Signal Step Response

The next step is further investigation into some of the signal dynamics and an attempt to identify any limiting or dominant factors. Figure 3.10, 3.11, and 3.12 show the various step responses when all 16 *sv* inputs are switched at once for different load conditions. Each figure has two plots, one for each of the two outputs. The effects of the cascode transistors were also investigated, and in each case, the solid line is the output with a functioning cascode transistor, and the dashed line shows the circuit output when the cascode transistor was pulled from its socket and the collector to emitter was replaced with a jumper. The results shown in these figures reveal a multitude of high order effects which are probably nonlinear with circuit origins that are difficult to identify completely. It should

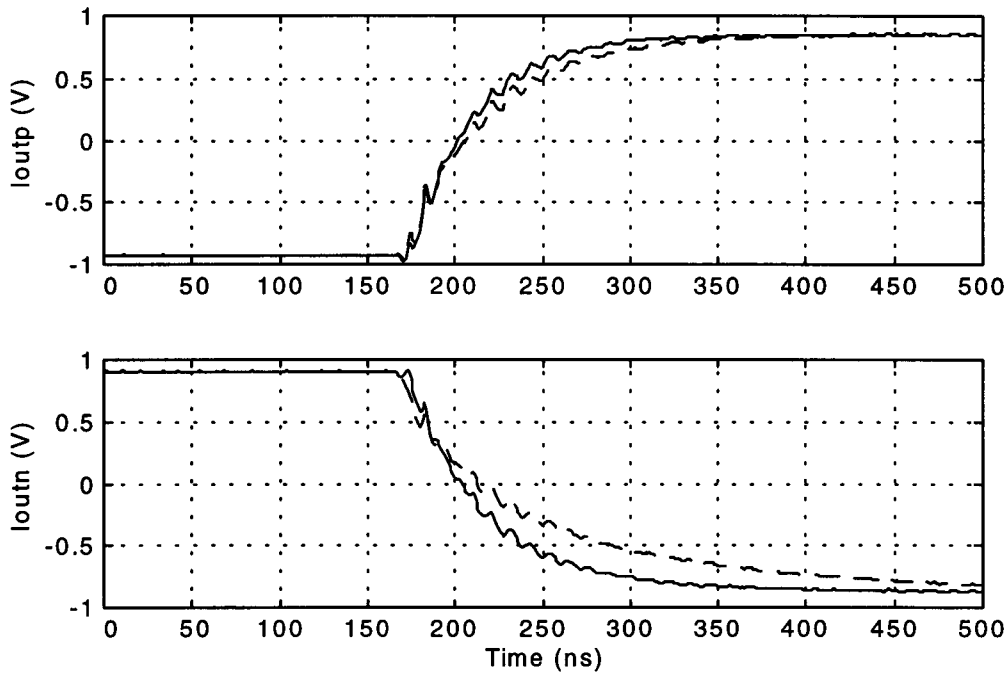


Figure 3.11:Step response of the ueDAC with *Rpu* = 1.1 kΩ. Dashed lines are the results without the cascode transistor.

also be noted that in order to zoom in on these signals, the scope had to be AC coupled, so the location of the zero-level on the voltage scale should be ignored. In reality, no negative voltages occur as the entire signal has a positive DC bias.

When presented with the complex signals arising from these step tests, we seek the dominant effects first, and then address higher order effects as necessary. One dominant feature is a roughly exponential decay as verified by overlaying a true exponential with $\tau = 15$ ns on the graph of Figure 3.10a using MATLAB.

Since increasing *Rpu* by a factor of 3 to 1.1 kΩ increases the time constant to about 50 ns (Figure 3.11) it is clear that the settling behavior is dominated by the RC time constant of the external load, rather that by the internal dynamics of the *ueDAC*. To test whether the capacitance is internal to the chip or external, a cascode transistor was used (the

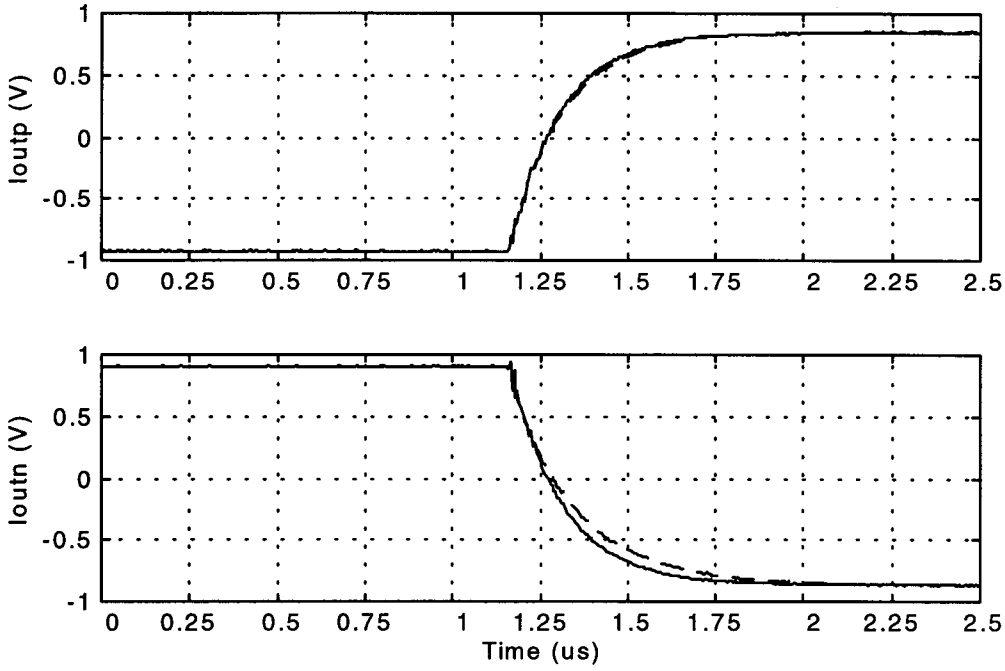


Figure 3.12:Step response of the *ueDAC* with *Rpu* = 1.1 kΩ, and C_L = 100 pF. Dashed lines are the results without the cascode transistor.

dashed curves in Figs. 3.10-3.12). In these cases the results show little difference with or without the cascode transistor, and from this we can conclude that while some capacitance is located prior to the transistors, and perhaps within the chip, the dominant capacitance is

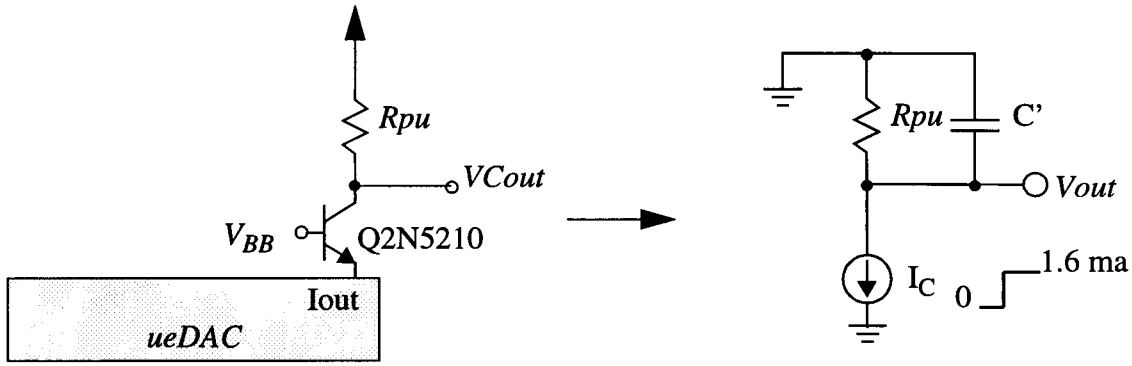


Figure 3.13:*ueDAC* output circuit and 1st order approximation assuming C' is the parasitic capacitance at the output node.

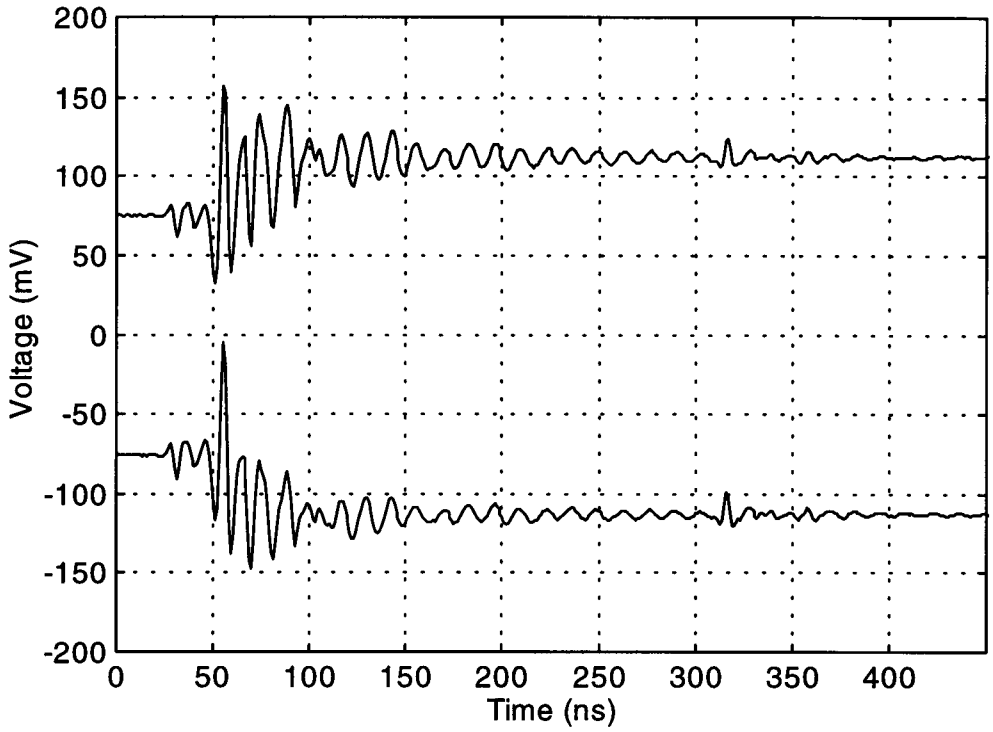


Figure 3.14:Small Signal Step Response of Ioutp and Ioutn due to one *sv* input being switched. (cascode transistors in place, *Rpu* = 371 Ω)

outside the chip. With no other load components on the board, this leads to the conclusion that the dominate pole consists of the parasitic board capacitance between the signal trace and the nearest low-impedance node, the ground plane. Since it now seems reasonable to attribute the dominant capacitance as outside the chip, Figure 3.13 shows a first-order model of the *ueDAC* output. The data of Figs. 3.10 and 3.11 indicate that C' is about 45 pF. As a further test, the results in Figure 3.12 were obtained when an additional load capacitance of 100 pF was added in parallel with *Rpu*. In this case, $\tau \cong Rpu(C' + 100pF)$. The time constant in Figure 3.12 is about 160 ns, and with *Rpu* = 1.1 kΩ, this also gives a C' of about 45 pF.

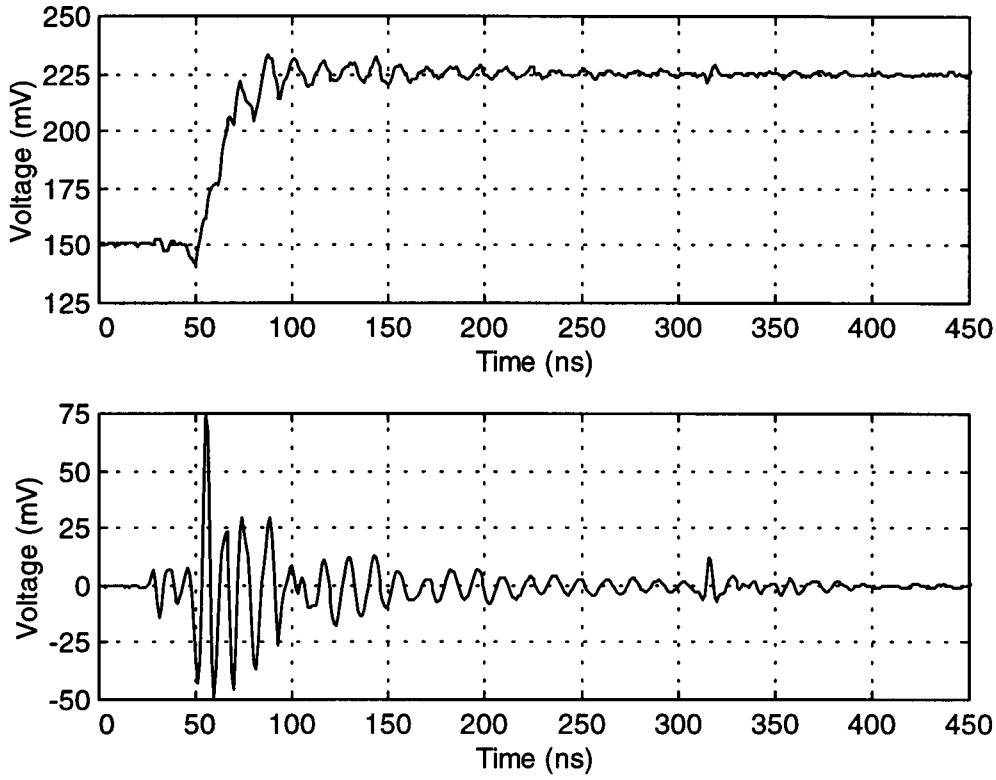


Figure 3.15:Top plot: (Ioutp-Ioutn), and bottom plot: (Ioutp+Ioutn)/2.

3.8 Small Signal Step Response

Figure 3.14 shows the small-signal step response of Ioutp and Ioutn due to one *sv* input being switched, while Figure 3.15 plots their average (common-mode component) and difference (differential-mode component). The results in Figure 3.15b show that the ringing in Figure 3.14 is largely common to both outputs, with the most severe portion of the waveform exhibiting ~100 MHz ringing over a 50 ns interval. This large common-mode component is suggestive of a significant amount of ground bounce internal to the IC.

3.9 Glitches

In [11] it was suggested that changing the voltage on B3 might minimize glitches. Testing showed that changing B3 has little to no effect on any observable switching glitches. However when B3 is set below about 2 V the outputs stop responding. Noting this, a value of 2.5 V is recommended.

3.10 Summary

The operation of the *ueDAC* was verified using both DC and AC tests. DC tests showed that the current sources have a standard deviation of 0.7% and an output resistance in excess of 10 MΩ. AC tests revealed that the current-source dynamics are dominated by the RC time constant of the external load. At *Rpu* = 371 Ω, the output has approximately 50 mV_{p-p} of 100 MHz ringing which lasts for about 50 ns.

With the basic operation of the *ueDAC* verified, the next step is to investigate its use in a mismatch shaping delta-sigma DAC system. Chapter 4 describes the schematic and board layout of the prototype circuit, details the modulator design and implementation, and presents the main experimental results of this work.

Chapter 4. Demonstration of Shaping

This chapter presents experimental results which show that the static nonlinearities in a multilevel DAC can be noise-shaped. For this demonstration we connect a 3rd order modulator implemented in a Xilinx field-programmable gate array to the *ueDAC*. The modulator design process is documented, the test set-up described, and lastly the measurements are presented.

4.1 Modulator Design and Simulation

Armed with a working chip and the Delta Sigma Toolbox [12], we are now ready to design a modulator suitable for driving the *ueDAC*. The design steps which were followed to implement the 3rd order 4-bit modulator are detailed below.

1. Create an input signal to be used for simulation and to be implemented in the final hardware.

It was desired to produce a 1/2-scale 16-bit sine wave. The logic implementing the modulator would be in 2's complement, so our input word would will have both positive and negative numbers. Realizing this signal was accomplished by generating the integer representation of a 16-bit sine wave in MATLAB. The frequency response of the truncated signal was verified to ensure that our signal was inherently more "pure" than our DAC's ideal performance, and the vector was saved to a file called "u2".

To accommodate the hardware implementation, a custom Perl script was used to convert "u2" into Intel Hex. Since our EPROMS were only 8-bits wide, the Perl script put the high and low bytes in separate files.

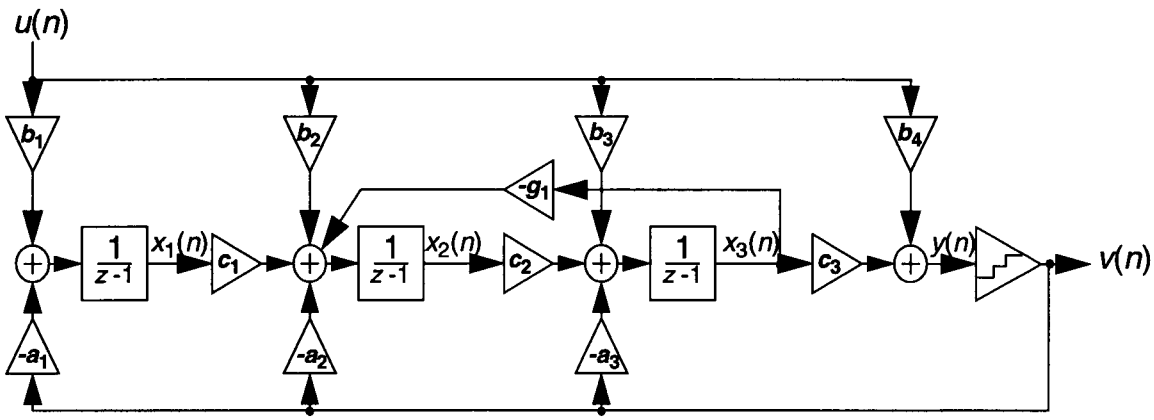


Figure 4.1: High level block diagram of the topology chosen for our modulator, the Cascade of Integrators Feedback Structure.

Next, the digital sine wave was programmed into a pair of 27C64 EPROMS using the EMP20 programmer. By clocking both of these EPROMS with the same signal, the desired 16-bit sine wave was produced.

2. Design and simulate the modulator in MATLAB.

Since we will be using the 16-element *ueDAC,* the modulator should be a 17-level design. Furthermore, it is desired that the theoretical SNR should be greater than about 96dB. Other than that, there are no hard design constraints except than the final modulator (and selection logic) must fit in the Xilinx 4005 FPGA.

The Delta Sigma Toolbox allows this task to be accomplished with remarkable ease. This thesis will not attempt to describe the Delta-Sigma Toolbox in full detail, however a few examples of its functionality pertaining to this modulator design will be given. The reader is encouraged to refer to [12].

The initial design work consisted of using the **synthesizeNTF** and **realizeNTF** functions. As the names imply, **synthesizeNTF** synthesizes a noise transfer function and **realizeNTF** converts a noise transfer function into coefficients for a specific structure.

Figure 4.1 shows the high level block diagram for the topology chosen in this design, the cascade of integrators feedback structure, and the following two lines of MATLAB code compute the coefficients.

```
>>H = synthesizeNTF(3,32,0,3);%Order=3,OSR=32,Opt=0,Hinf=3
>>[a, g, b, c] = realizeNTF(H, 'CIFB');
```

While these coefficients appear with ease, they are in no way practical for circuit implementation. Good design requires that several steps be followed as summarized below:

i) After the semi-infinite precision coefficients are obtained with **realizeNTF**, the first step is to simplify the final circuit implementation by modifying the coefficients at the expense of modulator performance. We iteratively simplify the coefficients and then simulate the modulator to determine if our desired SNR is still achieved. Amazingly, we find in this modulator that setting b_2, b_3, and $b_4 = 0$ increases the SNR by a small amount. Since setting b_2-b_4 to zero will greatly simplify the circuit implementation, we conclude that the effort spent in this step is well worth it.

ii) Next, we use the Delta-Sigma Toolbox to automatically perform scaling on the new coefficients determined above such that the simulated state maxima are all less than one with some margin. This step helps to determine the word width required in each block. As before, the modulator is simulated after scaling to make sure we still have our desired SNR.

iii) The final design step is to quantize the scaled coefficients into sums and differences of powers-of-2 to eliminate the need for costly hardware multipliers.

Powers-of-2 multiplies and divides can be realized by simple bit shifts, and generally come for free in the Xilinx. The design portion of this step is to experimentally determine how accurately we must quantize the coefficients while maintaining our desired SNR. This step degrades the modulator performance slightly and the trade offs are again simplicity for performance. It should be noted that this step is necessary in our design because the required hardware multipliers won't fit in the Xilinx 4005.

Table 4.1: Modulator coefficients and simulation results.

Coefficients	Modulator Coefficients, Cascade of Integrators Feedback Structure.	Simulated Peak SNR (dB)	Simulated state maxima
Semi-infinite	a = 0.4138, 1.5042, 1.9820 g = 0 b = 0.4138, 1.5042, 1.9820, 1.000 c = 1, 1, 1	101.7	3.1
Simplified	a = 0.4138, 1.5042, 1.9820 g = 0 b = 0.4138, 0, 0, 0 c = 1, 1, 1	102.9	3.1
Scaled	a = 0.0156, 0.0284, 0.0748 g = 0 b = 0.0156, 0, 0, 0 c = 0.5000, 2.0000, 26.4824	102.9	0.85
2^s-Quantized	a = 1/64,(1/32-1/256),(1/16+1/64) g = 0 b = 1/64, 0, 0, 0 c = 1/2, 2, (2^5-2^2)	100.7	0.85

Figure 4.2 displays the simulation results of the modulator designed thus far. These plots show the noise and signal transfer functions (the NTF and STF), simulated state maxima, a theoretical SNR curve, and finally the modulator's time-domain output with large and small input levels.

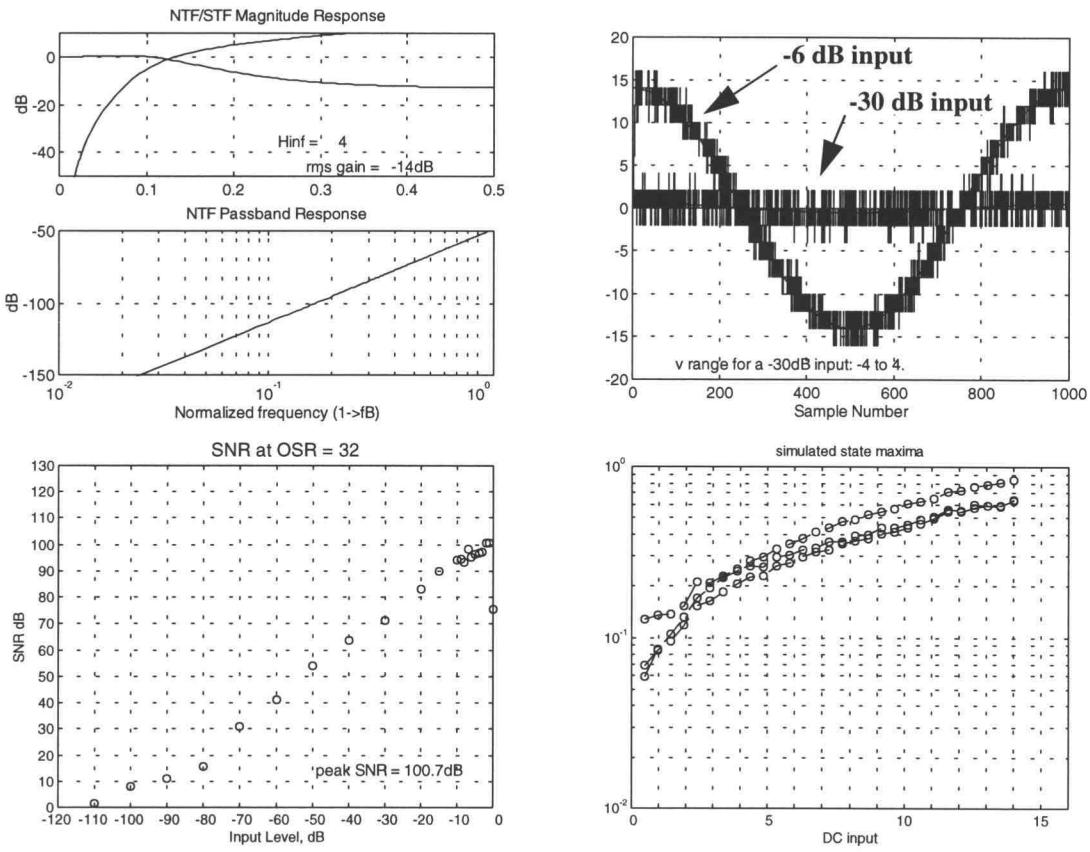


Figure 4.2: Simulated modulator behavior.

3. Circuit implementation.

Because we have made the effort to realize our multipliers as powers of 2, the bulk of the circuit requires only adders and delays with the appropriate bit shifts to obtain the coefficients. Good design will also require that the size of the adders be made as small as possible. In order to determine the minimum number of bits required in each state, the modulator was coded in C and verified to produce exactly the same output vector as the MATLAB simulation with the "u2" input signal. Examination of the C code in Appendix B will show that the number of bits to be used in each state is an assignment in the first few lines of the code. The iterative process was to start with more than enough bits required to represent each state, and then reduce the numbers until the modulator "broke". Appendix B

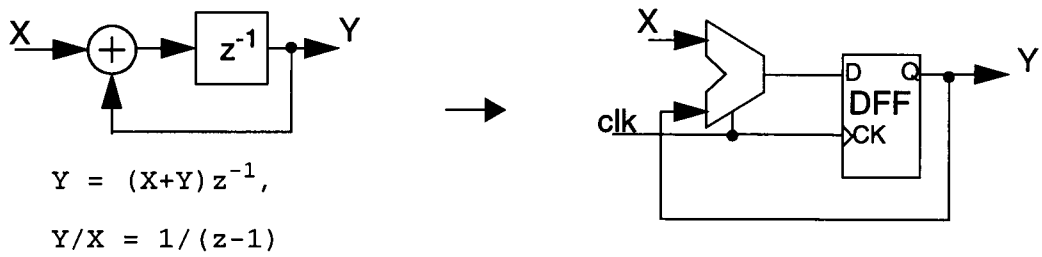


Figure 4.3: Circuit implementation of an integrator.

shows the schematic level implementation which was finally used. For background information, Figure 4.3 shows how the integrators are realized at the circuit level.

Finally, the last digital circuit required to complete the DAC is the element selection logic (ESL). Chapter 2 describes how the selection logic chooses the unit elements, so this will not be presented again here. Appendix B shows the logic circuits which realized the 0^{th}-order (thermometer decoder) and the 1^{st}-order selection logic.

4. Enter the modulator in a schematic capture program which can produce a net list suitable for Xilinx programming. In this case, Viewlogic was used.
5. And finally, using the "u2" input, verify with simulations that the ViewLogic schematic exactly matches the C code simulation.

As previously mentioned, these guidelines were followed and led to a functional modulator. Appendix B contains schematics for the modulator and the 1^{st} order selection logic along with the C code which simulates the modulator and a truth table which specifies the selection logic.

For review, we have simulated that a 3^{rd} order, 4-bit modulator suitable for practical circuit implementation should have an SNR of about 100 dB at an OSR of 32 with an ideal unit element DAC. Since we will be using the *ueDAC*, and the actual value of the 16 current sources are known, we can present another simulation of the more complete system. In

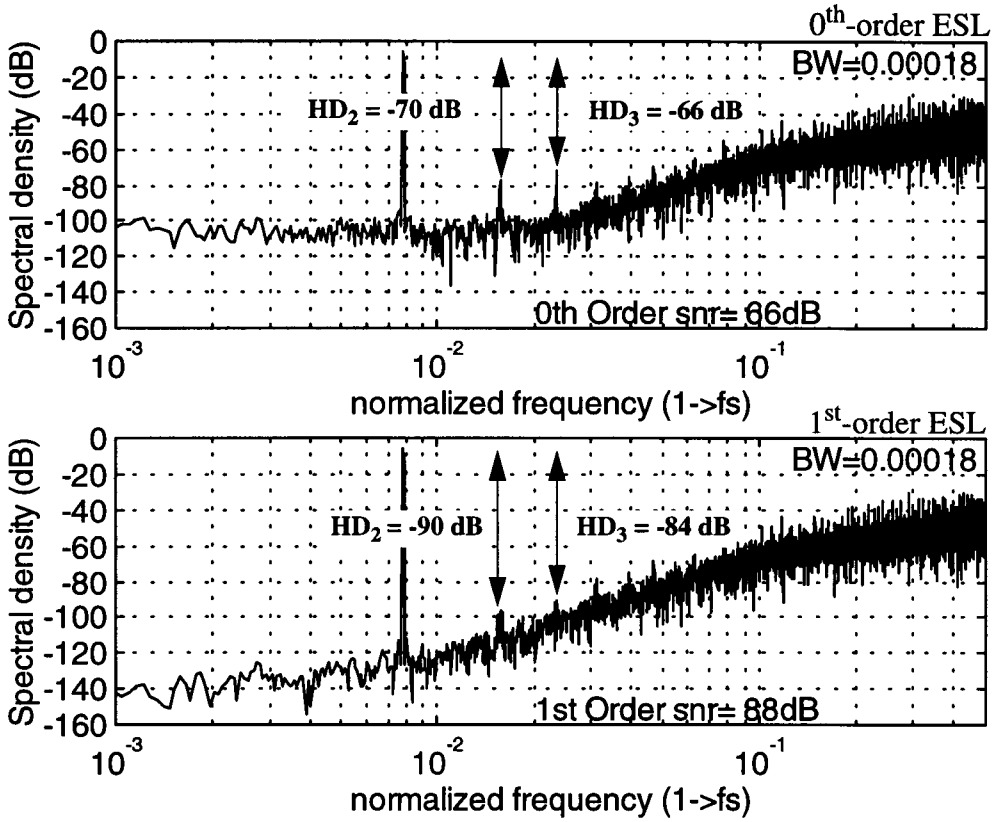


Figure 4.4: MSDAC simulation using the actual element values for the *ueDAC* chip tested.

Figure 4.4 we show the simulation results with the nonideal *ueDAC* values taken from the chip characterization data. Also, for demonstration purposes, Figure 4.5 shows the simulation results when one of the current sources is disabled (by setting sv_{10} to zero). In both figures the top plot presents the 0^{th}-order ESL and the bottom plot presents the 1^{st}-order ESL. Experimental results when sv_{10} is disconnected from the modulator and grounded will be shown in a later plot.

4.2 Test Board Design and Construction.

Designing a test board to drive the *ueDAC* is straightforward, but the importance of keeping the noise level as low as possible should be emphasized. Aside from that, the major

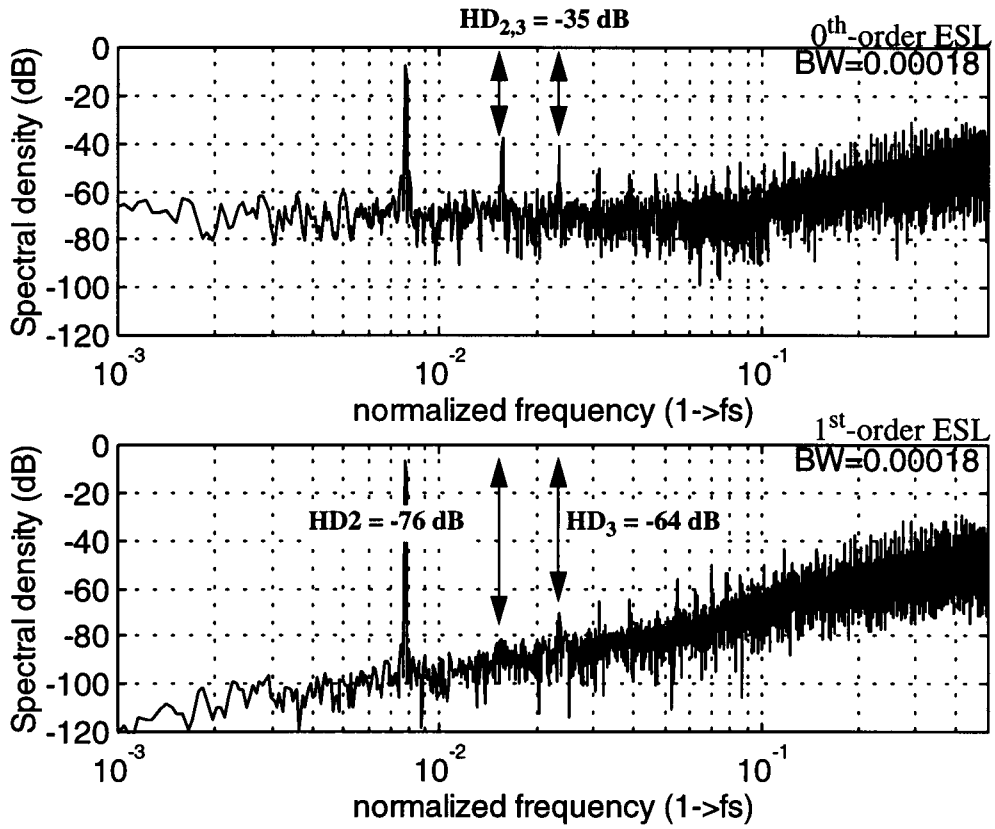


Figure 4.5: MSDAC simulation with sv_{10} "broken".

design considerations are how to realize the digital logic, what type of input signals are required, and to a lesser extent, versatility for other projects.

The test board which was constructed addressed the noise issue by using separate supplies and ground planes, decoupling capacitors, commercial voltage references for the bias voltages, and a layout which emphasized spacing between noisy digital signals and sensitive analog nodes. The placement of the *ueDAC* was also taken into consideration. Internally, the *ueDAC's* layout physically separates the analog and digital sections, and this is represented by its final placement on the test board. On a final note, a 1/4" wide no-route-zone of bare board was placed between the ground planes.

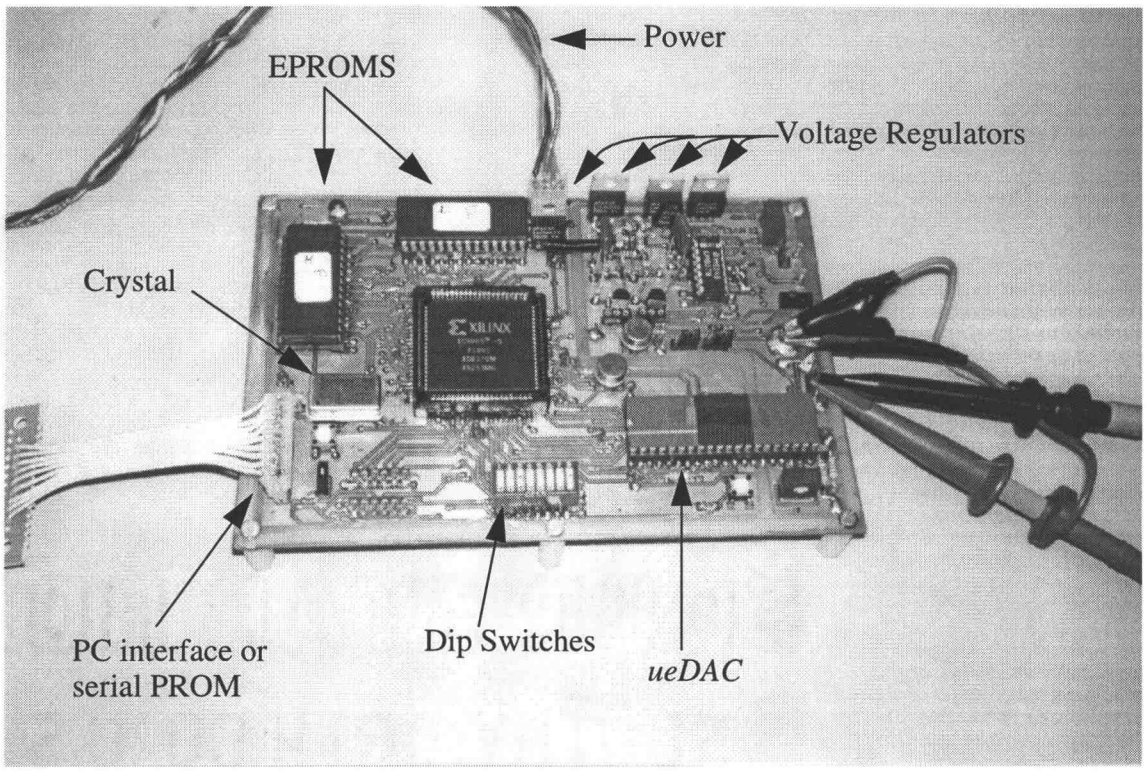


Figure 4.6: Photograph of the circuit board.

Two issues in this project were how to realize the large amount of digital logic needed, and how to generate a 16-bit digital sine wave as an input. Since building digital circuits and tearing them apart when changes are needed is time-consuming and error-prone, the solution was to store sine waves in a pair of 8-bit EPROMs and to use a Xilinx 4005 FPGA to hold all of the logic driving the *ueDAC*. This design allows us to easily change the input signal, and by using an FPGA we avoid reworking the hardware. Appendix B contains the complete circuit diagram of the test board, and Figure 4.6 shows a photograph of the final completed circuit board. The DIP switches were used to select the ESL order and clock speeds.

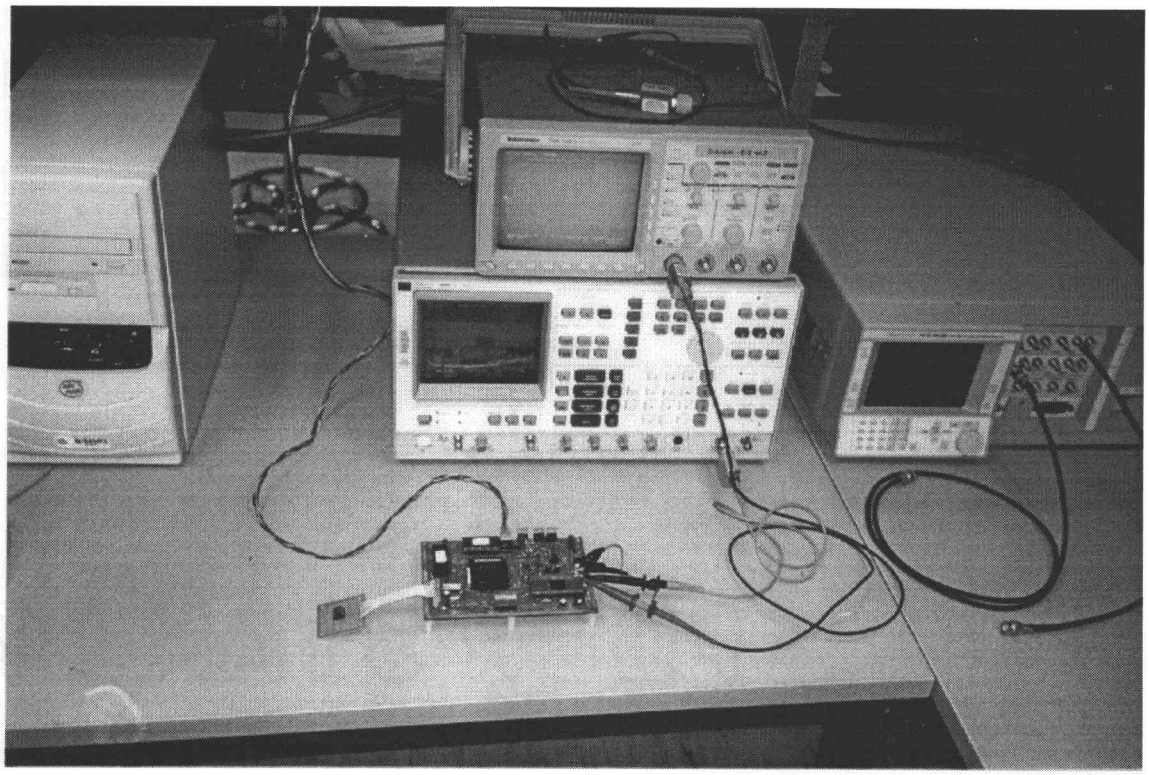


Figure 4.7: Photograph of the test setup and laboratory equipment.

4.3 Experimental Results

Figure 4.7 shows a photograph of the actual test setup for taking all of the experimental data. The lab equipment consists of a Tektronix TDS 420 scope, a Hewlett-Packard 3585B spectrum analyzer, and a PC to acquire data from either the scope or the spectrum analyzer via a GPIB port. Close inspection will reveal actual waveforms on both the spectrum analyzer and the scope.

In short, the DAC works. As a clear demonstration of noise shaping, Figure 4.8 presents the output spectrum up to $f_s/2$ on both a linear and logarithmic scale, and in Figure 4.9 we extend the scale to $2f_s$.

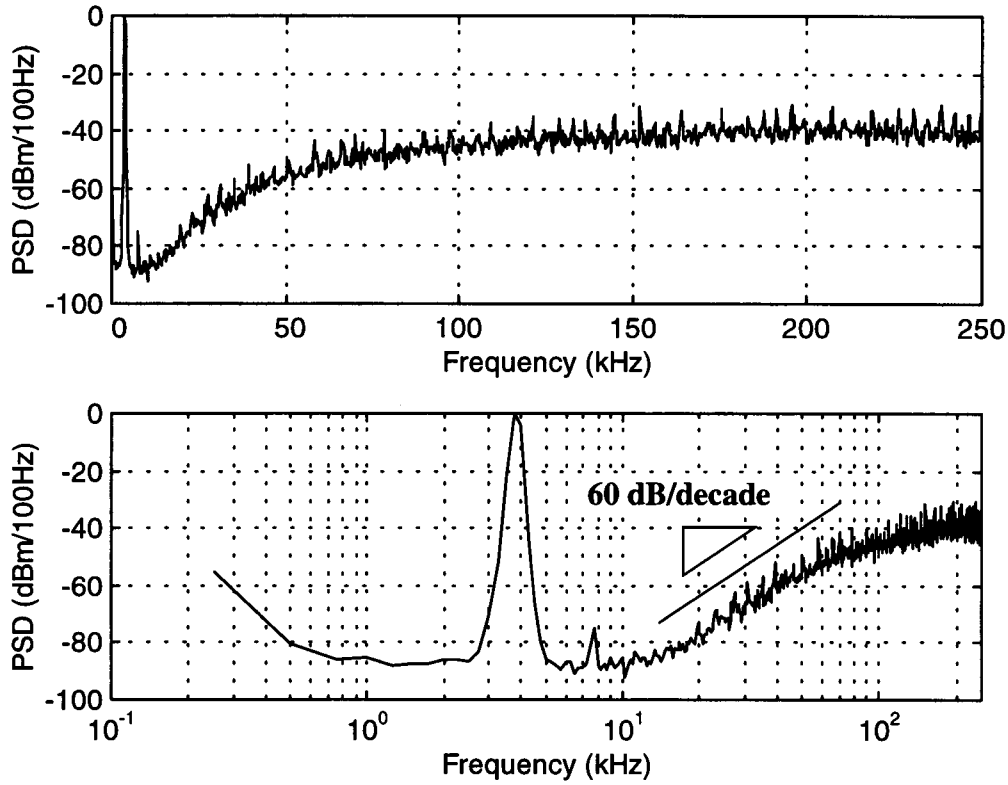


Figure 4.8: Output spectrum with f_s = 500 kHz, 0^{th} order ESL.

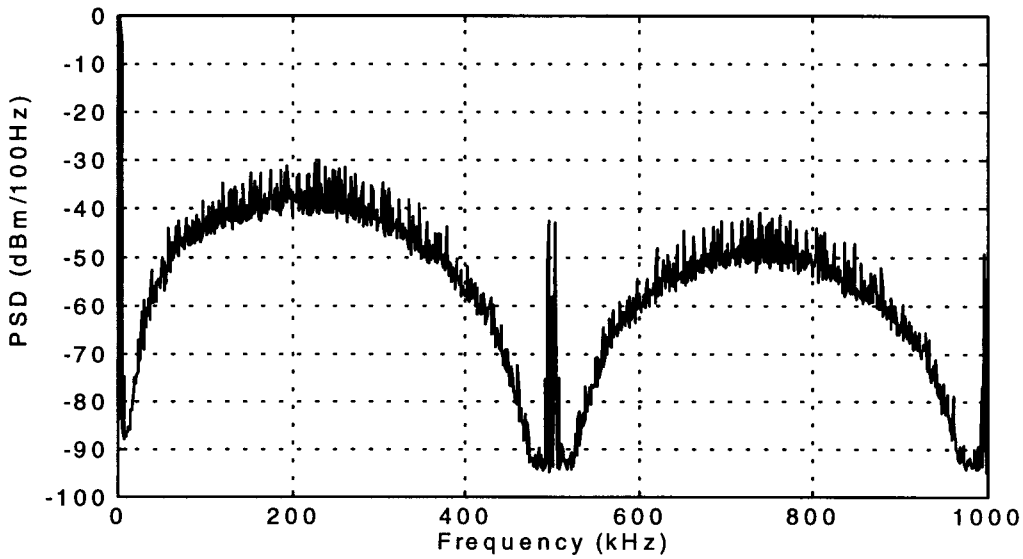


Figure 4.9: Output spectrum with f_s = 500 kHz.

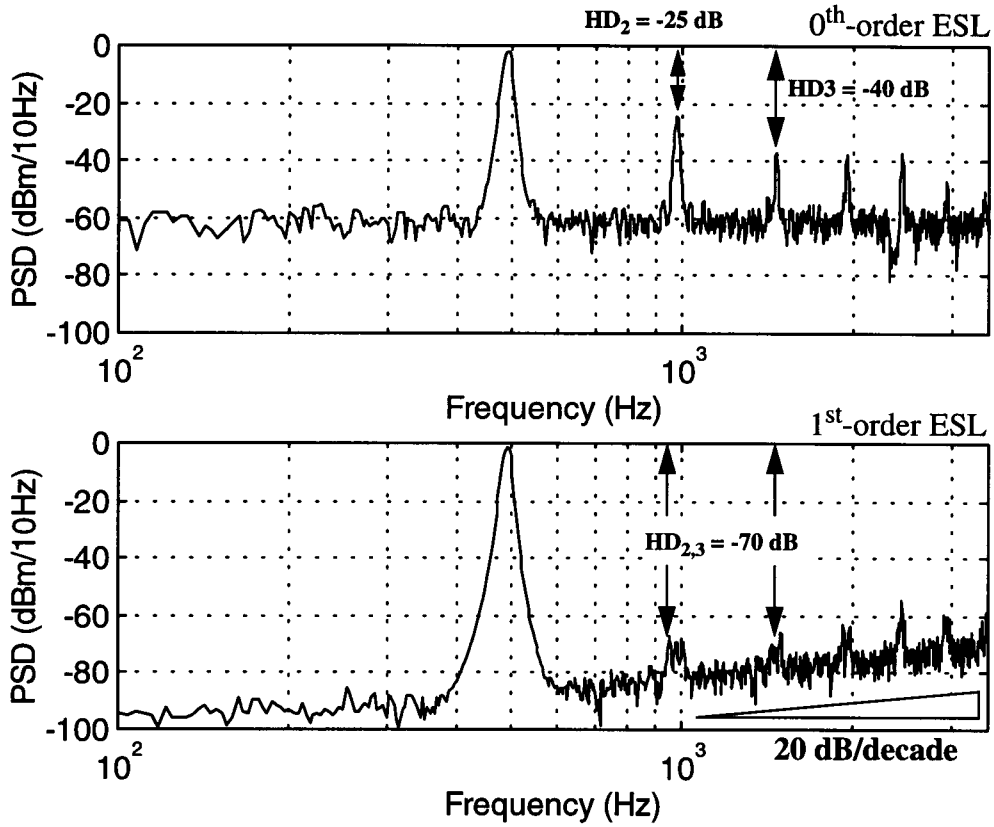


Figure 4.10:Measured data- Spectra for f_s = 62.5 kHz, sv_{10} disconnected from the selection logic and grounded.

Figure 4.8 shows that the spectrum exhibits a deep notch at low frequencies, with the 60 dB/decade slope characteristic of a triple zero at DC. Figure 4.9 shows the spectral images at f_s and $2f_s$ that one expects in a sampled-data system.

Theory predicts that 1st-order shaping improves the performance over 0th-order shaping, and in Figure 4.10 we have an exaggerated demonstration by artificially creating a 100% mismatch in one of the unit elements (by grounding sv_{10}). As shown in the top plot, disabling one of the sources destroys the quality of the output sine wave when mismatch-shaping is not used. (The simulations in Figure 4.5 predicted harmonic distortion at the -35 dB level; measurements show that HD_2 = -25 dB and HD_3 = -40 dB.) When mismatch-

shaping is turned on, the 2nd and 3rd order harmonic distortion improves to about -70 dB (simulations predicted -70 and -60 dB, respectively). In addition, we see from Figure 4.10b that the slope of the noise is consistent with the 20 dB/decade slope characteristic of first-order shaping. We therefore conclude that the mismatch-shaping logic is working.

Figures 4.11-4.18 show the low-frequency content of the output as the sampling rate is stepped in octave increments from f_s = 62.5 kHz to f_s = 8 MHz. In these figures the frequency range is from 0 to f_s/18 (corresponding to an OSR of 9), and the top plot shows the 0th-order shaping while the bottom plot shows the 1st-order shaping.

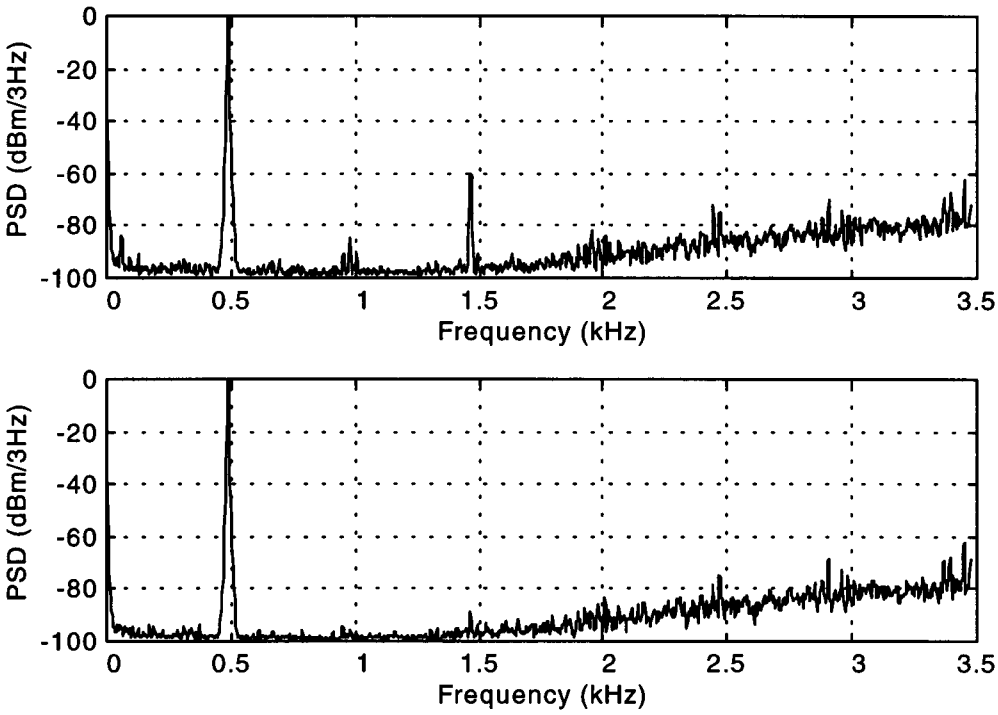


Figure 4.11:Measured data- Spectra for f_s = 62.5 kHz.

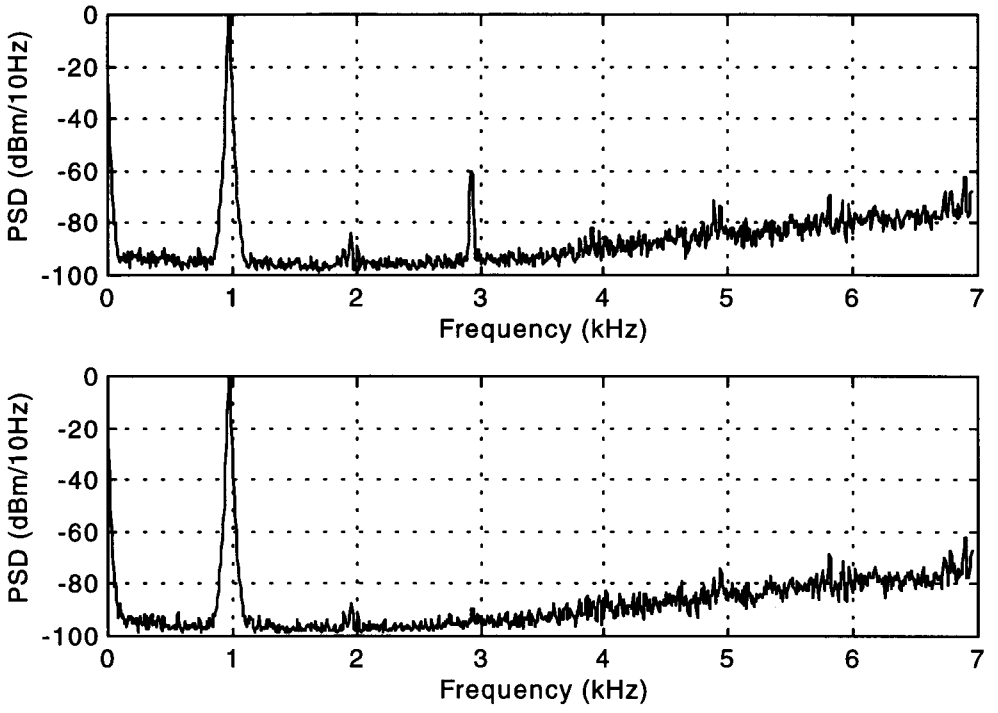


Figure 4.12:Measured data- Spectra for f_s = 125 kHz.

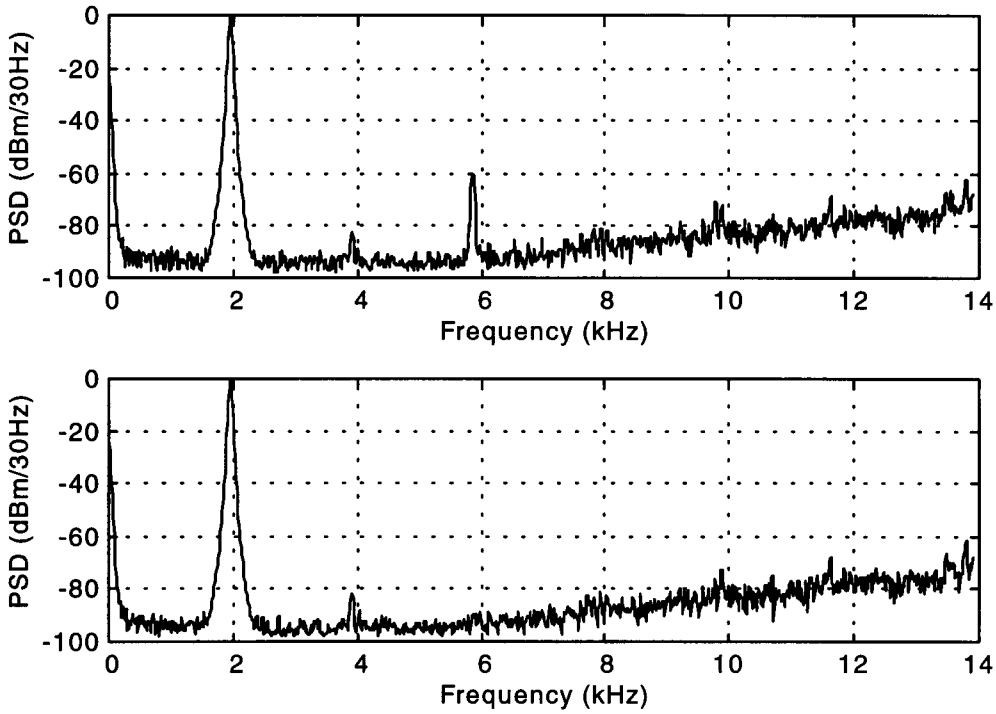


Figure 4.13:Measured data- Spectra for f_s = 250 kHz.

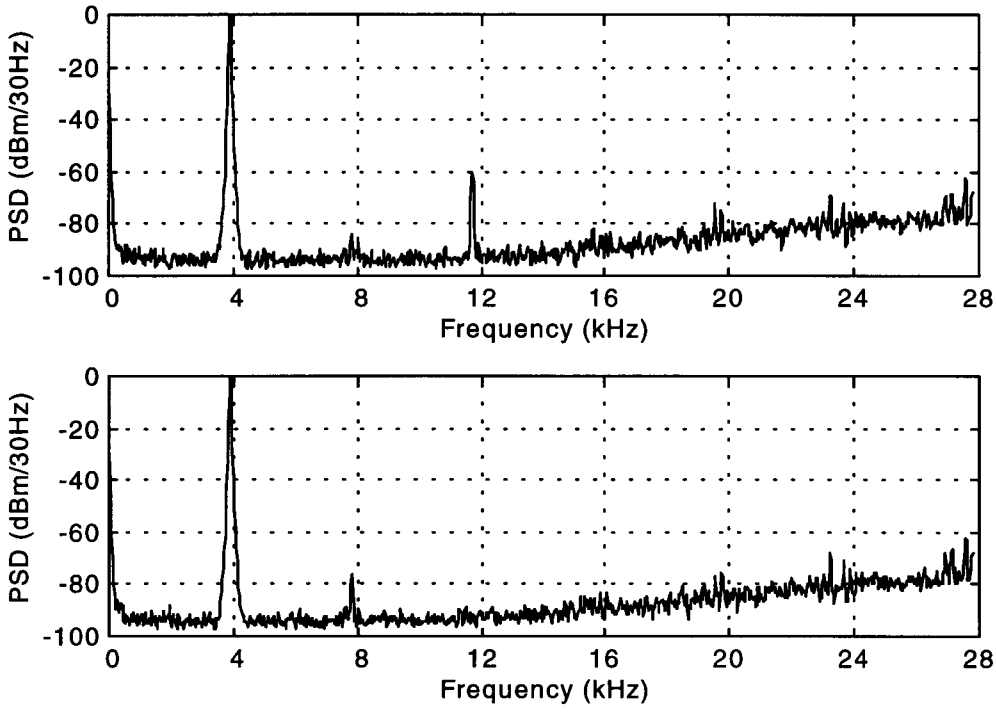


Figure 4.14:Measured data- Spectra for f_s = 500 kHz.

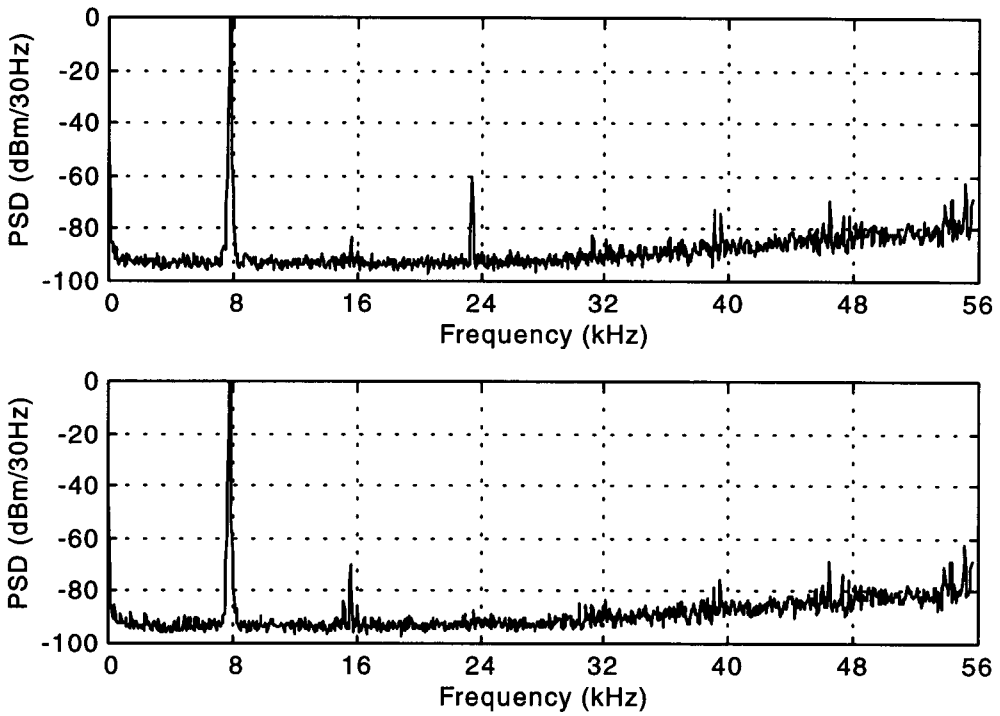


Figure 4.15:Measured data- Spectra for f_s = 1 MHz.

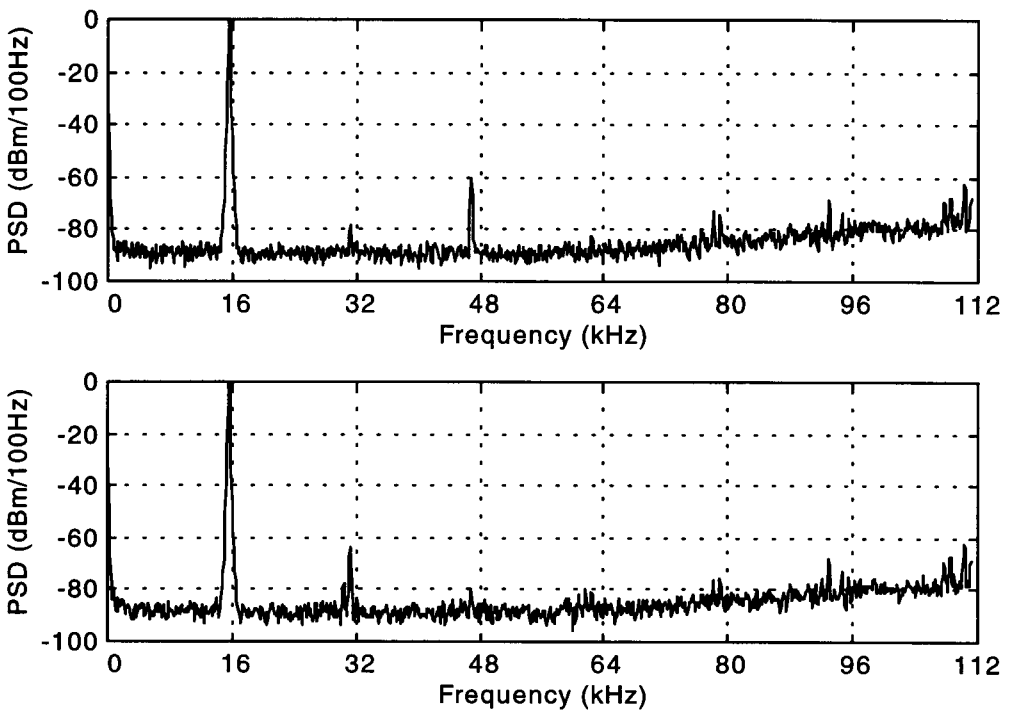


Figure 4.16:Measured data- Spectra for f_s = 2 MHz.

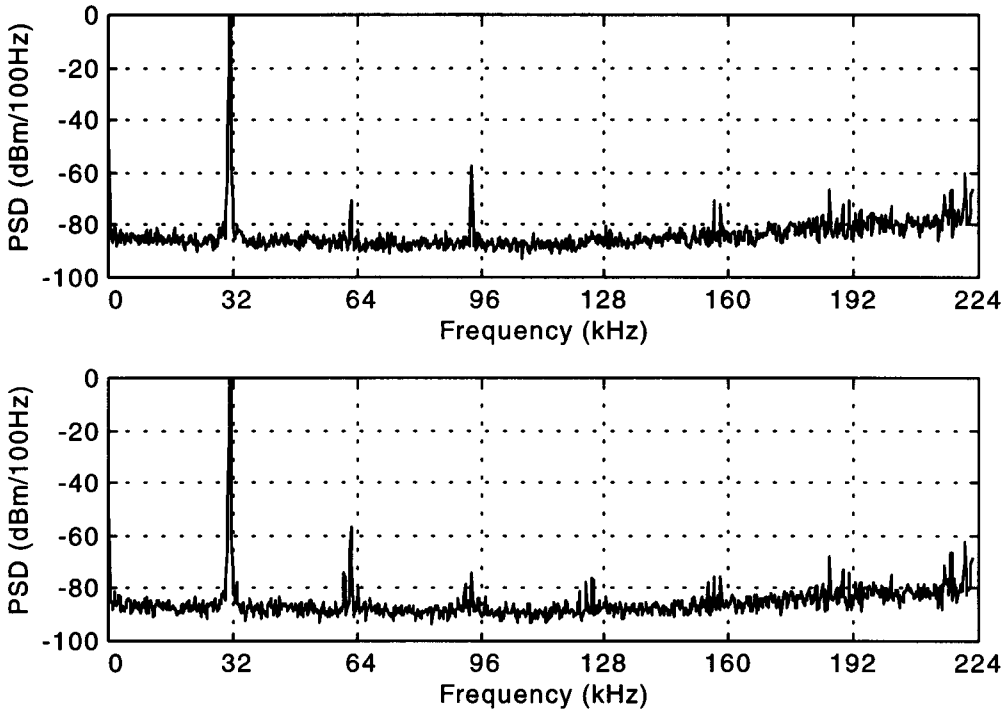


Figure 4.17:Measured data- Spectra for $f_s = 4$ MHz.

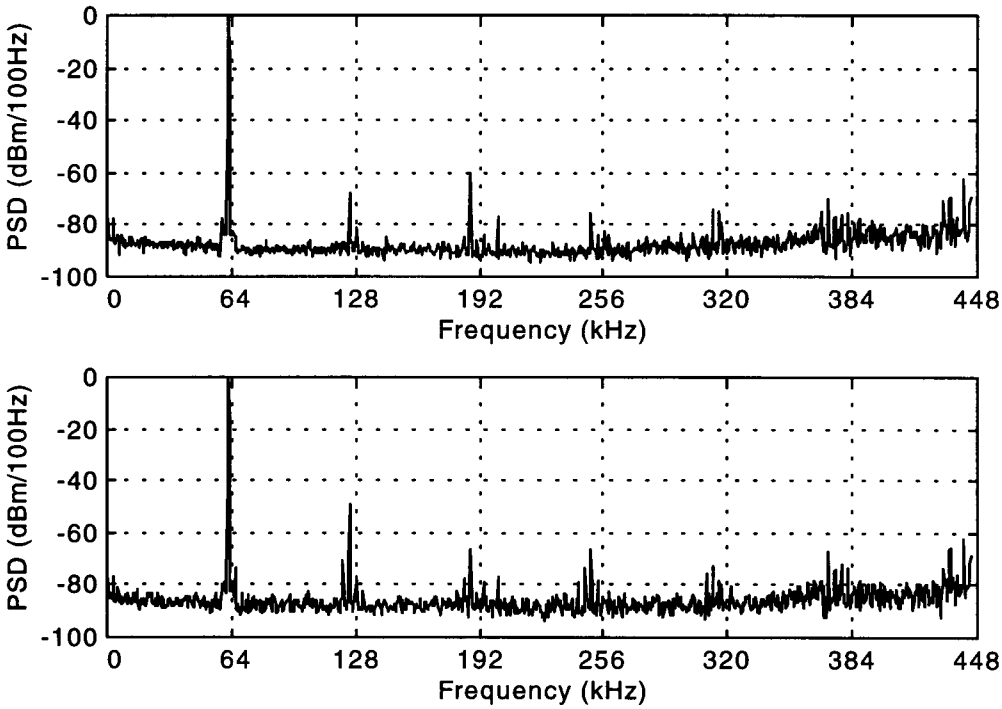


Figure 4.18:Measured data- Spectra for $f_s = 8$ MHz.

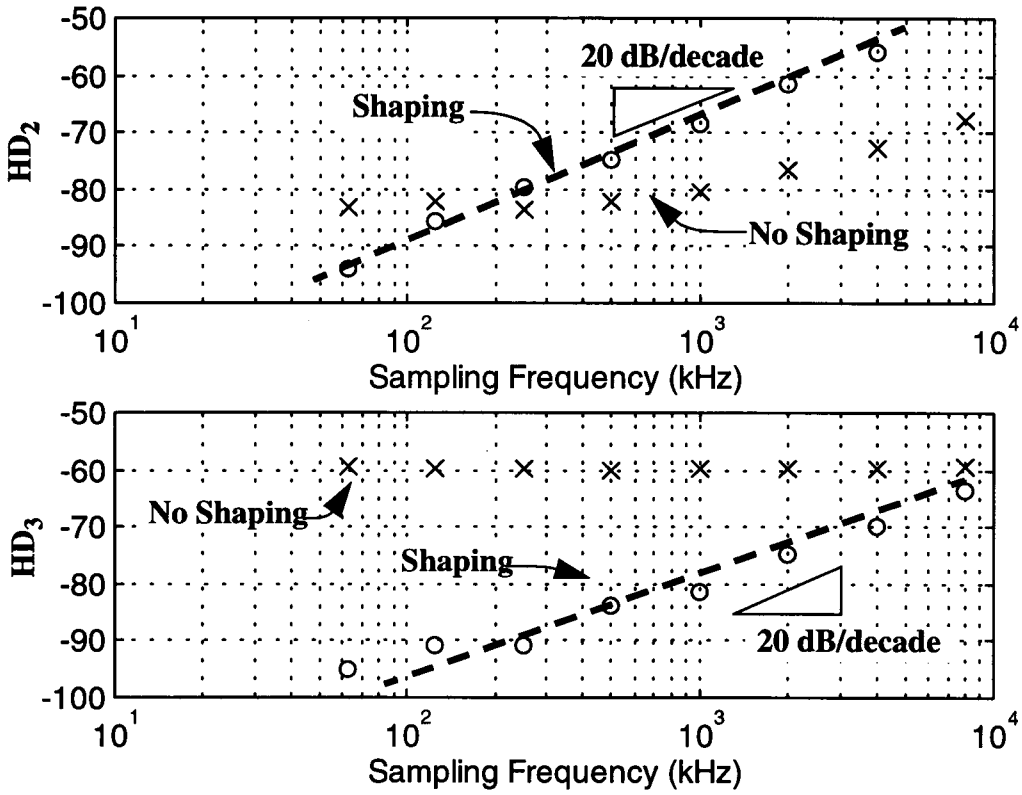


Figure 4.19:Measured harmonic distortion as a function of sampling rate.

Ideally, the circuit performance would be independent of f_s. Unfortunately, the measurements show that the performance of the DAC deteriorates as f_s increases. A particularly disturbing discovery is that even though the 3rd harmonic is generally suppressed when shaping is used, the 2nd harmonic generally increases. This is especially visible in Figures 4.14-4.16, where f_s ranges from 500 kHz to 2 MHz. Figure 4.19 summarizes the data from Figures 4.11-4.17 by plotting the measured 2nd and 3rd harmonic distortion as a function of clock frequency. In what follows, explanations for the observed behavior will be offered.

4.4 Discussion of Results

The results of the static characterization presented in Section 3.4 were used in the system simulations of Section 4.1 to predict the performance of the DAC system before and after the application of mismatch-shaping. Figure 4.4a predicted that when plain thermometer-coding is used, element mismatch will result in a second-harmonic distortion of HD_2 = -70 dB and a third-harmonic distortion of HD_3 = -66 dB. Figure 4.4b predicts that mismatch-shaping will improve these figures to HD_2' = -90 dB and HD_3' = -84 dB, respectively. The goal of this section is to reconcile the measurement results of Figure 4.19 with the predictions of Figure 4.4.

Since the discrete-time system simulations do not take into account effects related to the continuous-time nature of the waveforms present in the real circuit, the predicted distortion figures are not dependent on the sampling frequency. However, Figure 4.19 clearly shows frequency-dependence in the distortion, especially when mismatch-shaping is used. When mismatch-shaping is not used, HD_3 shows almost no frequency-dependence, while HD_2 is only weakly frequency-dependent. When mismatch-shaping is used, both HD_2' and HD_3' exhibit a 20 dB per decade slope. In other words, over the range of sampling frequencies tested, HD_2' and HD_3' are proportional to f_s. We will discuss each of these observations in turn, starting with those associated with the non-shaped case.

The predicted value of HD_3 is -70 dB, 10 dB better than the observed value of -60 dB. The fact that the actual HD_3 is independent of frequency indicates that, at this level of distortion, dynamic effects are swamped by mismatch-induced distortion. Thus, one would expect very good agreement between discrete-time simulations of the characterized system and actual measurements. One possible source of the discrepancy is the values measured for the current sources. MATLAB experiments indicate that by applying random

perturbations (having an rms value of 0.3μA) to the current sources, the simulated HD_3 can be increased to the observed value. Although this level of error is high considering the precision of the Fluke 87 used to make the measurements, it is not beyond the realm of possibility.

The second curve to consider is that associated with HD_2. Here, low-frequency measurements indicate that HD_2 is about 14 dB *better* than the predicted value. At first glance, this would appear to be a very unlikely occurrence (measurements better than theory), but is in fact easy to account for. First, recall that the data plotted in Figure 3.6 exhibited almost perfect even symmetry. In fact, the data of Table 3.3 indicate that the difference between elements on either side of the line of symmetry is less than 0.1μA (the precision of the Fluke 87). Second, note that an exactly even-symmetric distribution of current-sources would eliminate the even components in a power-series representation of the integral non-linearity of the DAC. Thus, for a test sine wave centered precisely on the point of symmetry, all even harmonics will disappear. In light of these observations, it is not surprising that the symmetry actually present in the current sources is better than the static characterization indicated and thus the measurements of HD_2 are better than predicted. MATLAB simulations were used to verify these statements, and to determine the degree by which the symmetry must be improved in order to achieve HD_2 = -82 dB. Reducing the asymmetry by a factor of 5 (a change of only $0.07\mu A_{rms}$ to the measured values) is sufficient to account for the discrepancy between simulation and measurement. The rise in HD_2 with frequency for f_s > 1MHz will be briefly discussed in conjunction with the distortion associated with the shaped case.

The last part of this discussion focuses on why HD_2' and HD_3' are proportional to f_s. The second harmonic is particularly worrisome, not just because HD_2' is bigger than HD_3', but also because HD_2' is generally bigger than HD_2, i.e. at most values of f_s, mismatch-shaping worsens the second harmonic distortion. For these reasons, HD_2' will be treated first.

One possible explanation for the rise in HD_2' with frequency is related to the fact that each component of the selection vector becomes a delta-sigma sequence encoding the desired sine wave when mismatch-shaping is applied. Thus, 16 high-amplitude digital signals containing spike energy at the output frequency (f_1) all converge on the IC when mismatch-shaping is used. Any coupling between these signals and the reference current will result in an f_1 component on the reference. Since the reference is a multiplicative input to the system, a $2f_1$ component will therefore be created at the output. In order for the amplitude of the resultant second harmonic to be proportional to frequency, the degree of coupling must also be proportional to frequency, suggesting a capacitive coupling to the reference input. To check this possibility, the voltage at the IB pin was monitored using the spectrum analyzer and the existence of an f_1 component was indeed observed. To verify that this component is truly responsible for the observed harmonic content, two tests were made:

i) The IB pin was bypassed with a large capacitor.

ii) The amplitude of the f_1 component was measured and compared with the results of Figure 4.19.

The addition of a bypass capacitor on the IB pin was observed to eliminate the f_1 component of the (external) voltage on the IB pin, but unfortunately this did not reduce HD_2'. Thus, either the existence of an f_1 component is not responsible for HD_2', or

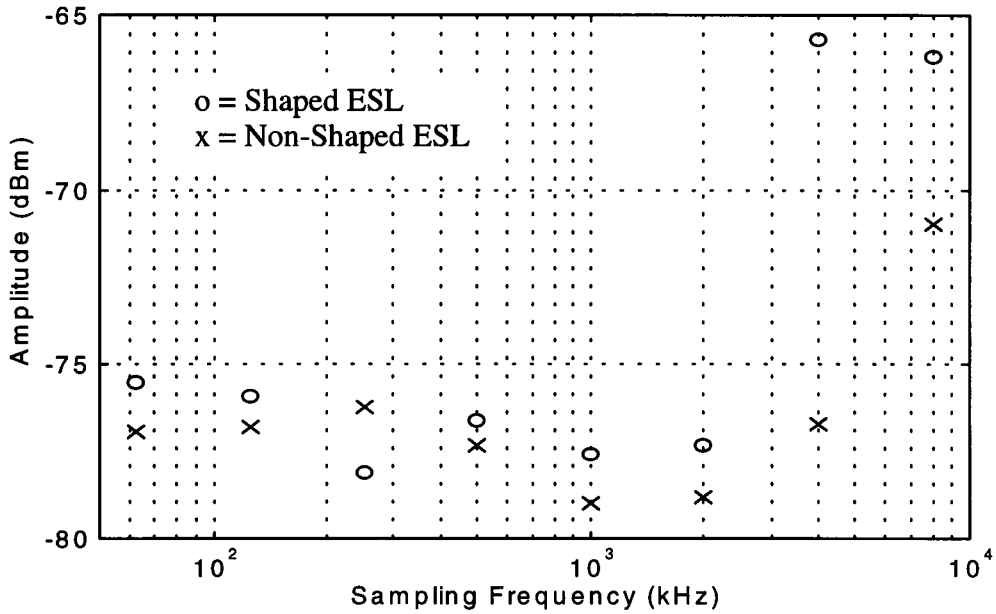


Figure 4.20:Magnitude of the fundamental on IB.

eliminating the f_1 component on the voltage external to the IC does not affect f_1 component on the reference current internal to the IC. To provide more information, the amplitude of the f_1 component is plotted in Figure 4.20. We can see immediately from this Figure that the variation in the f_1 component on IB in the shaped case does not track the HD_2' curve. Specifically, HD_2' varies by 40 dB over the frequency range tested whereas the f_1 component on IB shows only a variation of about 10 dB. It follows that corruption of the reference by the signal is not the limiting factor in the shaped case.

Observe, however, that the non-shaped data in Figure 4.20 curve upward at approximately f_s = 3 MHz, very close to where HD_2 rises in Figure 4.19. Furthermore, the variation in HD_2 (roughly 15 dB) is comparable to that seen for the IB pin (about 10 dB). To see if corruption of the reference by the signal is a plausible source of the second-harmonic distortion in the non-shaped case, a calculation is in order.

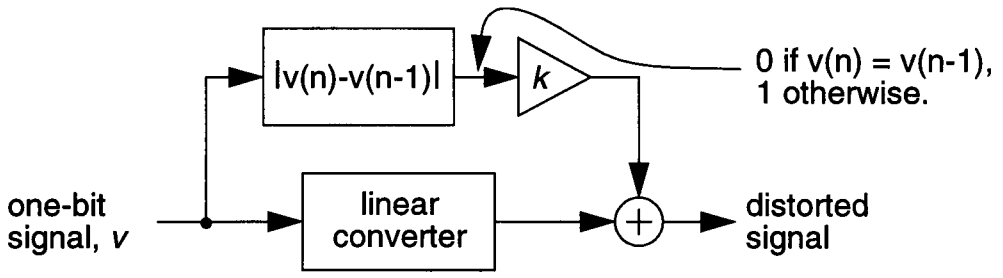


Figure 4.21:Modeling the distortion caused by mismatched rise and fall times.

At f_s = 4 MHz, the signal on IB reported by the spectrum analyzer is -77 dBm and Eq. (4.1) shows the relationship between dBm and the actual voltage present.

$$\text{dBm}=10\log_{10}\left(\frac{V^2}{R}\times 1000\right), \text{ where R = 50 } \Omega. \tag{4.1}$$

Solving Eq. (4.1) for V gives 31.6μV_{rms} which translates to about 5 nA_{rms} of corruption on IB. This leads to a predicted value of HD_2 = -89 dB, which is 16 dB lower than the observed value of -73 dB. Although the calculated result is within an order of magnitude (and thus is "in the ball-park") the discrepancy is large enough that it is not possible to definitively conclude that corruption of the reference is the source of the observed HD_2.

Since corruption of the reference does not account for the observed HD_2', another explanation must be offered. It is well known[13] that the process of converting a discrete-time signal into a continuous-time waveform is a non-linear operation unless the rise and fall characteristics of the waveform are symmetric. For binary signals, the distortion caused by the conversion from discrete to continuous-time can be modeled by the system shown in Figure 4.21. In this figure, the upper path adds an error signal to the output whenever there is a transition on the one-bit input. As the figure suggests, this error signal is derived in a non-linear manner from the input. If one assumes that the rise and fall time-constants (or time delays) of an individual current source differ by an amount $\Delta\tau$, the gain k of the

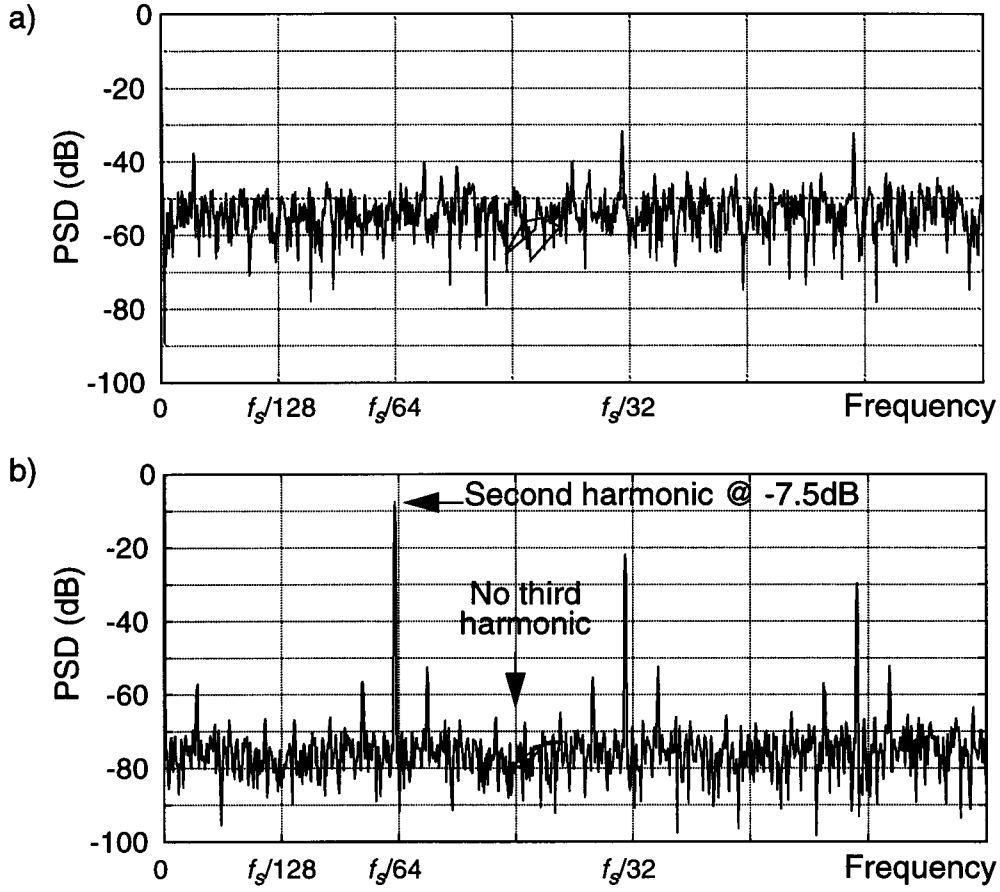


Figure 4.22:Spectra of $\|\Delta sv\|_1$ for a) non-shaped and b) shaped selection vectors.

nonlinear path relative to the linear path is then $(\Delta\tau)/T$, where $T = 1/f_s$. It is the fact that k is proportional to f_s that leads us to now consider mismatched rise and fall times as a possible cause of the observed distortion.

Figure 4.22 displays the spectra of $\sum_{i=1}^{M} |sv_i(n) - sv_i(n-1)|$, for both non-shaped and shaped sv signals. As this figure shows, there is a strong second harmonic present when a shaped sv vector is considered, but not when an non-shaped vector is used. This is consistent with the observation that shaping worsens HD_2. Correlating the spectrum of Figure 4.22b with the data of Figure 4.19a, we find that $\Delta\tau = 0.5$ ns is sufficient to account for the observed HD_2'. Since the rise and fall times of the waveform shown in

Fig. 3.10 are in the range of 15-20 ns (with considerable amounts of glitch energy) and since 0.5 ns is approximately the delay of an inverter implemented in the 1.2 μm technology used by the IC, it is entirely plausible that mismatched rise and fall characteristics are responsible for the observed HD_2'.

Unfortunately, since the spectrum of Figure 4.22b does not contain a third harmonic, mismatched rise/fall characteristics of individual elements explains neither the existence nor the form of HD_3'. To account for the observed HD_3', a mechanism whereby higher-order harmonics can be generated needs to be proposed. Such a mechanism was in fact touched upon in Chapter 3, namely nonlinear step dynamics. As with mismatched rise/fall characteristics, the effects of nonlinear step dynamics are more severe at high clock frequencies, rising at a rate of 20 dB per decade increase in f_s. However, unlike the effect of mismatched rise/fall characteristics on binary signals, nonlinear step dynamics can generate harmonics of any order. By analogy with the discussion of HD_2', a timing variation on the order of 0.2 ns is all that is needed to yield the observed levels of distortion.

4.5 Summary

This chapter presented the main results of the thesis. The design and implementation with a Xilinx FPGA of a third-order multibit digital modulator were described. The modulator was connected to the *ueDAC* chip described in Chapter 3 and the resulting delta-sigma DAC system was subjected to a variety of tests. The tests demonstrated that mismatch-shaping can be effective in reducing the distortion caused by element mismatch at low clock speeds. Using mismatch-shaping at a sampling frequency of f_s = 62.5 kHz, a ~500 Hz sine wave was produced which had harmonics more than 90 dB below the fundamental; without mismatch-shaping the third harmonic was only 60 dB below the fundamental. It was observed that the DAC's performance drops in proportion to f_s, and non-linear step dynamics are a likely culprit.

Chapter 5. Conclusions

5.1 Summary

Chapter 2 provided the reader with background information about delta-sigma modulation, starting with the lowly first-order single-bit modulator and progressing to advanced multibit modulators. The advantages of multibit DACs were highlighted and theory which suggested that the nonlinearity errors in a multibit DAC made with unit elements could be noise-shaped was presented. The theory is new enough that experimental verification has not yet been published in the open literature. It is the author's opinion that all theoretical developments, regardless of their simplicity or complexity, need to be verified with experiments. Only through such demonstrations are effects ignored by the theory brought to light and only when all such effects are understood can meaningful directions for further work be identified.

Chapter 3 focused on characterizing the 16-element *ueDAC*, thereby providing the information necessary for designing a demonstration system and for evaluating the test results. In this chapter, many important characteristics of the *ueDAC* were documented, from the pin-out to the voltage range and output resistance of a current source, to the matching of the current sources, and finally to the dynamics associated with the switching of the current sources.

The main contributions of the thesis are contained in Chapter 4. This chapter described the implementation of a digital modulator and included predictions of system performance based on the element values obtained experimentally in Chapter 3. The test board design and construction were detailed, and by deliberately disabling one of the unit elements in the *ueDAC*, a clear demonstration of the effectiveness of mismatch-shaping was made. With all elements active, mismatch-shaping was shown to reduce the harmonic

distortion for $f_s < 125$ kHz; harmonic distortion below -90 dB was observed for $f_s = 62.5$ kHz. It was further observed that when mismatch-shaping is used, the harmonics are proportional to f_s. Dynamic effects (asymmetric rise/fall characteristics and nonlinear step dynamics) were proposed as the cause of the observed behavior and correlated with the measurements using the data gathered earlier in Chapter 3.

5.2 Future Work

Despite the considerable amount of testing conducted during this work, more and better measurements could still be made. The actual output resistance of the current sources could be determined (as opposed to the lower bound presented in Chapter 3). A more accurate measurement of the current source mismatch could be carried out, thereby allowing the discrete-time simulations to better match the measurements. Also, the test set-up could be improved by taking advantage of the differential output of the *ueDAC*, thereby removing common-mode glitches and eliminating even-order harmonics. To allow more detailed characterization, the test set-up could also incorporate a variable-amplitude, variable-frequency digital sine wave source, as opposed to the half-scale, period-128 sine wave used throughout this document. The cause of the common-mode glitches could be tracked down and, if possible, eliminated. Finally, and perhaps most importantly, the SNR of the DAC could be measured. It should be noted here, if nowhere else, that the observed noise floor was low enough that for OSR > 16 the noise of the spectrum analyzer dominated the measurements.

As often happens in the course of verifying a new theory, several surprises were observed that have lead to suggesting possible refinements to the scheme originally envisioned. Since mismatch-shaping makes every component of the selection vector contain a strong tone at the test frequency, corruption of the reference and the ensuing second-harmonic distortion are virtually guaranteed. This effect can be mitigated simply by

scrambling the ***sv*** signals with a pseudo-random bit stream. Since mismatch-shaping causes elements to turn on and off more frequently, mismatch-shaping accentuates the detrimental effects of non-ideal step dynamics. When these detrimental effects dominate, a more intelligent element selection algorithm should take into account the need to switch elements less often. Lastly, the effects of interaction between simultaneously-switched elements might be lessened by staggering the switching times.

Bibliography

[1] R. L. Geiger, P. E. Allen and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits*, pp. 642-6472, McGraw-Hill Publishing Company, 1990.

[2] J. C. Candy and G. C. Temes, "Oversampling methods for A/D and D/A conversion," in *Oversampling Delta-Sigma Converters*, J. C. Candy and G. C. Temes, Eds. New York: IEEE Press, 1991.

[3] R. M. Gray, "Oversampled sigma-delta modulation," *IEEE Transactions on Communications*, vol. 35, no. 5, pp. 481-489, May 1987.

[4] M. W. Hauser, "Principles of oversampling A/D conversion,", *Journal of the Audio Engineering Society,* vol. 39, no. 1/2, pp. 3-26, January/February 1991.

[5] R. T. Baird and T. S. Fiez, "Improved ΔΣ DAC linearity using data weighted averaging", *Proceedings of the 1995 IEEE International Symposium on Circuits and Systems*,vol. 1, pp. 13-16, April 30-May 2, 1995, Seattle.

[6] R. T. Baird and T. S. Fiez, "Linearity enhancement of multibit ΔΣ A/D and D/A converters using data weighted averaging", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, pp.753-762, December 1995.

[7] M. H. Adams and C. Toumazou, "A novel architecture for reducing the sensitivity of mulitbit sigma-delta ADCs and DAC linearity", *Proceedings of the 1995 IEEE International Symposium on Circuits and Systems*, pp.17-20, April 30-May 3, 1995, Seattle.

[8] R. Schreier and B. Zhang, "Noise-shaped multi-bit D/A convertor employing unit elements", *IEE Electronics Letters*, Vol. 31,No. 20, pp.1712-1713, Sept. 28, 1995.

[9] W. L. Lee, "A novel higher order interpolative modulator topology for high resolution oversampling A/D converters," Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1987.

[10] A. V. Oppenheim, *Signals and Systems*, Prentice-Hall, 1983.

[11] J. da Silva, "Multi-Bit DAC with Noise-Shaped Element Mismatch", OSU internal report, October 1995

[12] R. Schreier, "Delta-sigma toolbox for MATLAB," version 2.0, ftp://next242.ece.orst.edu/pub/delsig.tar.Z, January 1997.

[13] R. W. Adams, "Design and implementation of an audio 18-bit analog-to-digital converter using oversampling techniques," *Journal of the Audio Engineering Society*, vol. 34, pp. 153-166, March 1986.

[14] L. Risbo "FPGA based 32 times oversampling 8th-order sigma-delta audio DAC," *Proc. 96th AES Convention,* preprint #3808, February 1994.

[15] J. C. Candy, "A use of limit cycle oscillations to obtain robust analog-to-digital converters," *IEEE Transactions on Communications*, vol. 22, no. 3, pp. 298-305, March 1974.

[16] The Xilinx, *The programmable logic data book*, San Jose, California, 1994.

[17] B. Zhang, "Delta-Sigma Modulators Employing Continuous-Time Circuits and Mismatch-Shaping DACs", *Ph.D. thesis, Oregon State University,* 1996.

[18] R. Schreier and B. Zhang, "Noise-shaped multi-bit D/A convertor employing unit elements", *IEE Electronics Letters*, Vol. 31,No. 20, pp.1712-1713, Sept. 28, 1995.

[19] H. Q. Lin, J. B. Silva, B. Zhang and R. Schreier, "Multi-bit DAC with noise-shaped element mismatch", Accepted to *1996 IEEE International Symposium on Circuits and Systems*, Atlanta, 1996.

Appendices

Appendix A -- Static matching of the unit elements in the *ueDAC*, tabular data

Table 1: Tabular Data Shown in Figure 3.7.

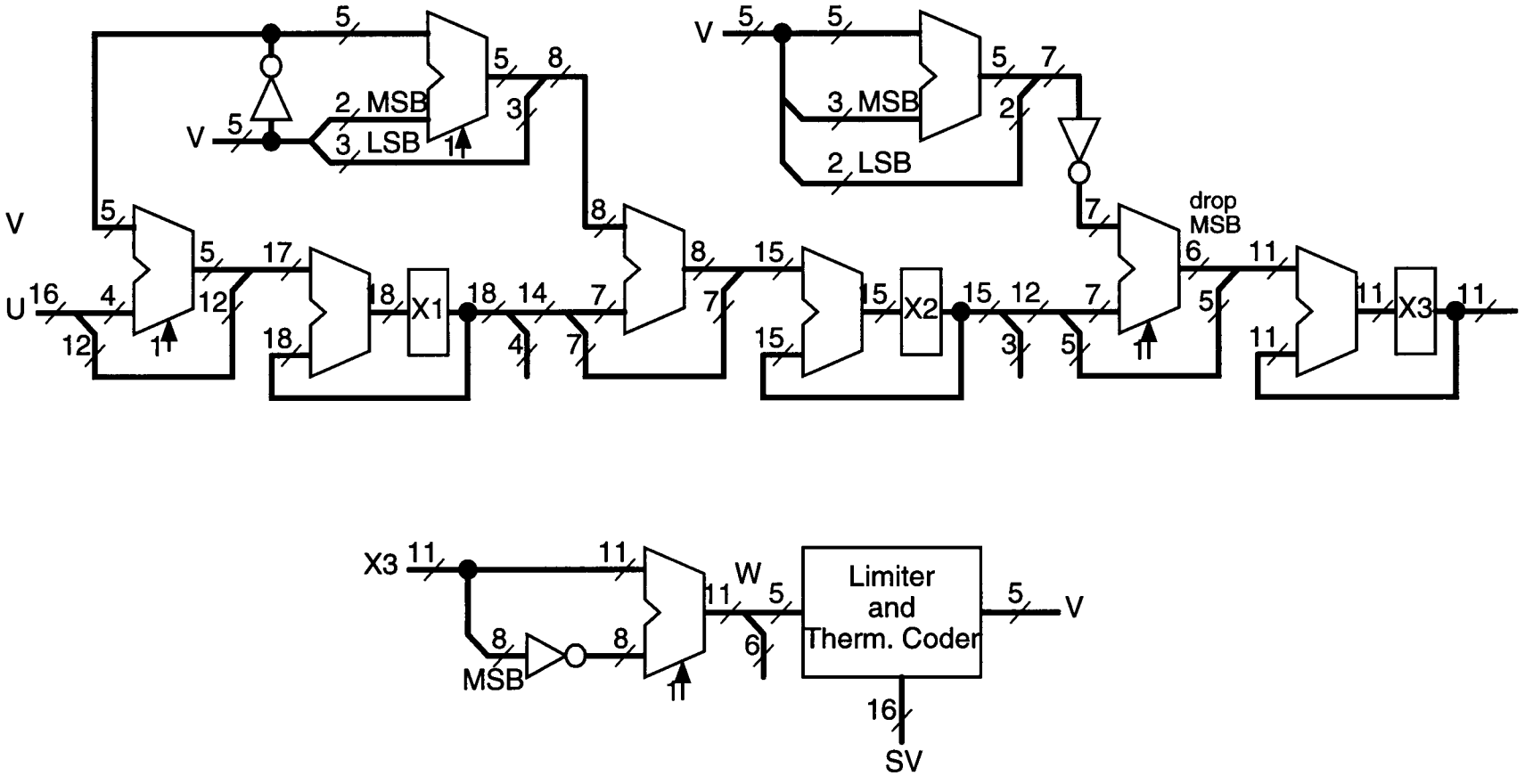
Testing IN SV8 = 0 Volts All others = 5 Volts		Testing IP SV8 = 5 Volts All others = 0 Volts		Testing IP All sv inputs at 5 Volts		Testing IN All sv inputs at 0 Volts	
Id, (μA)	*Vout*, (V)	Id, (μA)	*Vout*, (V)	Id, (μA)	*Vout*, (V)	Id, (μA)	*Vout*, (V)
4.2	0.807	9.0	1.091	17.21	0.139	6.51	0.745
12.9	0.833	20.1	1.101	45.69	0.144	64.0	0.756
21.4	0.86	37.1	1.116	155.8	0.164	125.3	0.768
41.4	0.928	46.5	1.125	219.2	0.176	228.0	0.789
51.3	0.964	55.5	1.136	336.5	0.200	303.5	0.804
60.8	1.003	69.2	1.158	429.1	0.220	478.0	0.841
71.3	1.049	85.9	1.193	511.1	0.239	563.1	0.860
81.3	1.101	95.3	1.233	668.0	0.278	695.3	0.891
88.5	1.145	100.8	1.341	755.5	0.303	735.0	0.901
100.0	1.285	100.9	1.397	833.6	0.327	1046.2	0.983
101.1	1.389	101.1*	1.615*	1114.5	0.438	1176.2	1.024
101.3*	1.642*	101.2	3.195	1254.0	0.513	1213.4	1.036
101.4	2.081	101.3	4.15	1318.9	0.555	1317.7	1.077
101.5	3.37	101.4*	5.02*	1492.8	0.700	1416.5	1.121
101.6	4.09	103.4	5.60	1571.7	0.807	1576.1	1.234
101.7*	5.06*	112.5	5.64	1624.1	1.004	1615.2	1.306
109.9	5.63			1625.0	1.020	1620.4	1.329
140.0	5.66			1630.4	1.150	1624.0	1.350
				1630.4*	2.094*	1626.5*	2.455*
				1630.5*	3.112*	1626.7*	5.00*
				1630.5	5.01	1630.9	5.61
				1638.2	5.62	1725.0	5.7
				1712.2	5.69		

* Indicates data used for output impedance calculations in Ch. 3

Appendix B -- Implementation of the Modulator

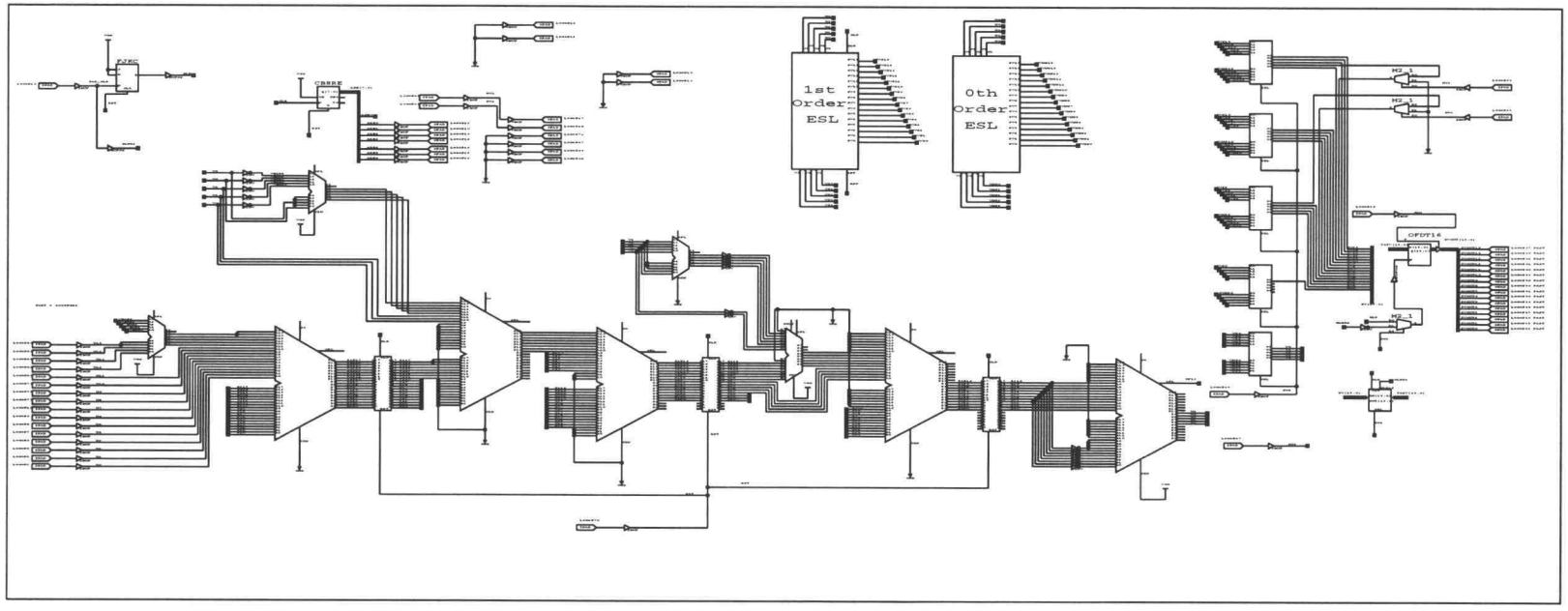
```
/* A C simulation of the modulator portion of the MSDAC */

#include <stdio.h>
main(){
int u, v=0;
int x1=0, x2=0, x3=0;/* state variables */
int b1=18, b2=15, b3=11; /* number of bits in each state (including sign
bit) */
int x1lim=1<<(b1-1), x2lim=1<<(b2-1), x3lim=1<<(b3-1);
int M1=(1<<b1)-1, M2=(1<<b2)-1, M3=(1<<b3)-1;

while(scanf("%d", &u)>0){
#ifdef TEST
 /* Make this simulation match MATLAB, where u is in [-16,16]. */
 u = u<<11;
#endif
 /* a1=1./64, a2=1./32-1./256, a3=1./16+1./64; */
 x3 = x3 + (x2>>(b2-b3-1)) - (v<<(b3-4)) - (v<<(b3-6));
 x2 = x2 + (x1>>(b1-b2+1)) - (v<<(b2-5)) + (v<<(b2-8));
 x1 = x1 + u - (v<<(b1-6));

#ifdef TEST
 /* Print the output and the scaled states in floating point format. */
 printf("%6d %8.4f %8.4f %8.4f\n", 2*v, (double)x1/x1lim, (double)x2/
x2lim, (double)x3/x3lim);
#else
#ifdef HEX
 /* Print the input, the states and the unscaled output, in
hexadecimal.*/
 printf("%04x %05x %04x %03x %02x\n", u, x1&M1, x2&M2, x3&M3, v&0x1f);
#else
 /* Print the output. */
 printf("%d\n", 2*v );
#endif
#endif
v = (x3 - (x3>>3) /* 28 = 2^5-2^2 */
#ifdef TEST
+ (1<<(b3-6))
#endif
          ) >> (b3-5);     /* -8:1:8 */
 if(v>8) /* Overflow can occur in the quantizer. */
         v=8;
 else if(v<-8)
         v=-8;
 if( abs(x1) > x1lim ){
         fprintf( stderr, "Error. Overflow in state 1.\n");
         exit(-1);
         }
 if( abs(x2) > x2lim ){
         fprintf( stderr, "Error. Overflow in state 2.\n");
         exit(-1);
         }
 if( abs(x3) > x3lim ){
         fprintf( stderr, "Error. Overflow in state 3.\n");
         exit(-1);
         }
 }
exit(0)
```

I. Modulator Implementation

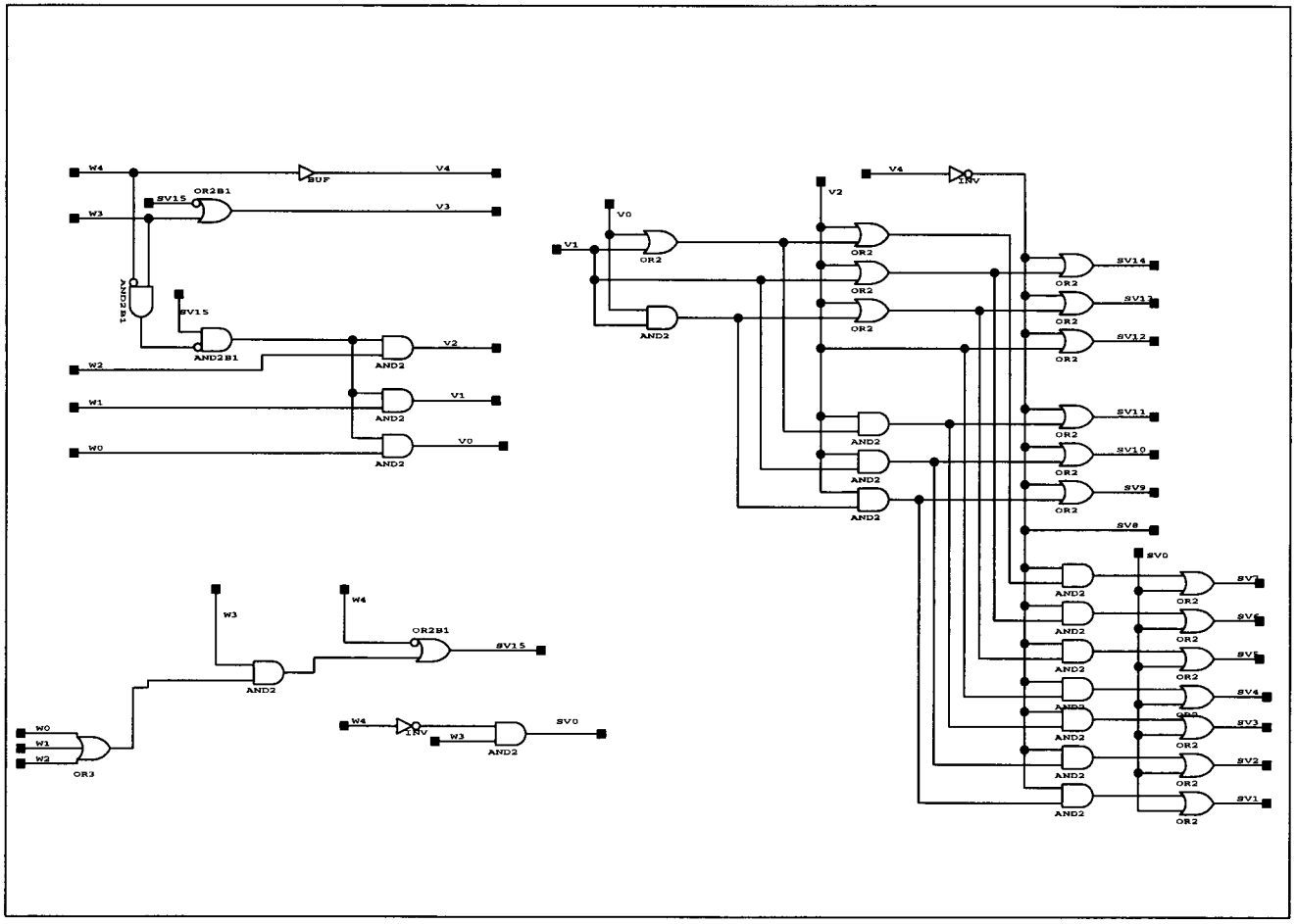


II. Truth table for the Limiter and Thermometer Coder

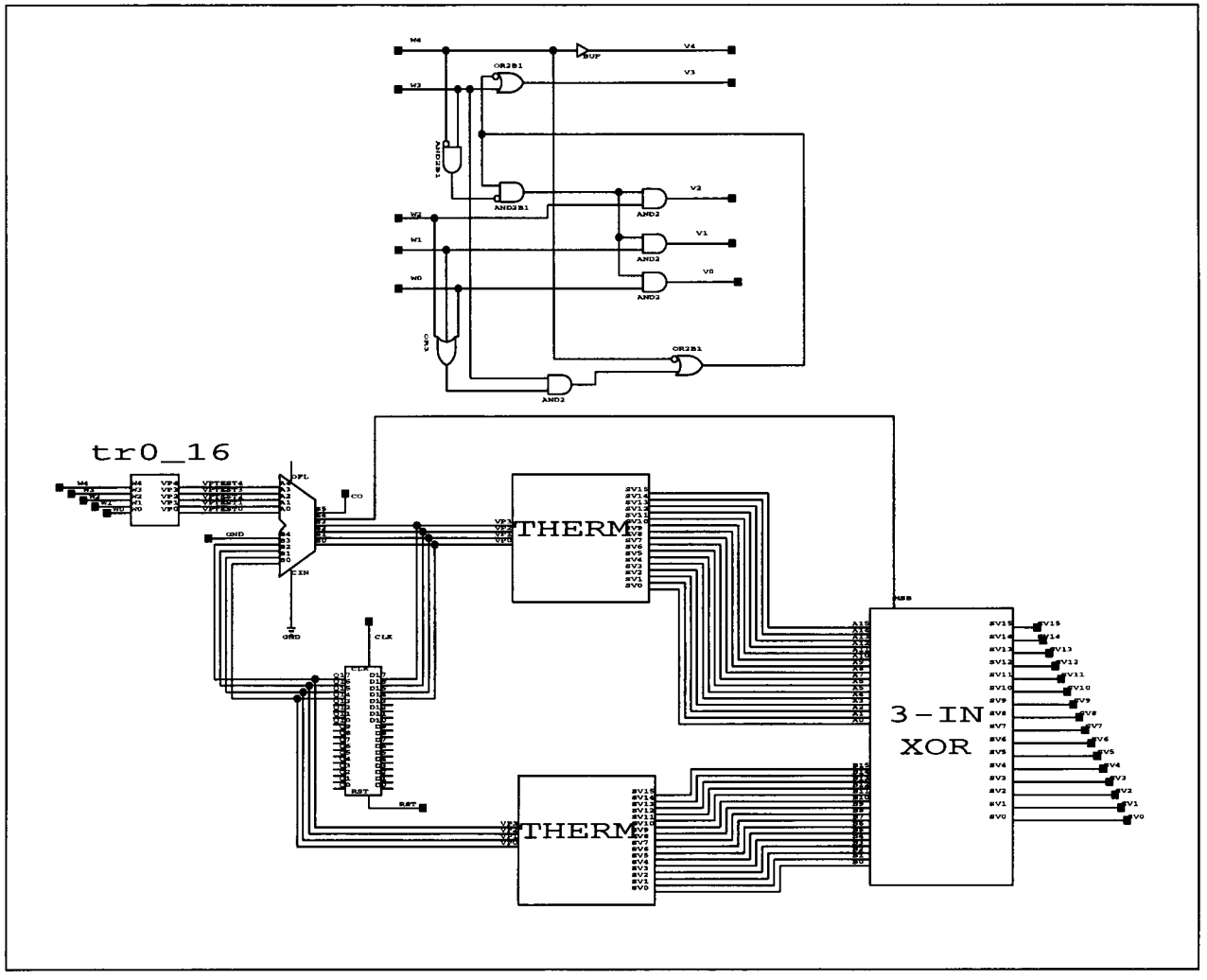
W4	W3	W2	W1	W0	V	SV0	SV1	SV2	SV3	SV4	SV5	SV6	SV7	SV8	SV9	SV10	SV11	SV12	SV13	SV14	SV15
0	1	X	X	X	8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	7	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	0	6	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	1	5	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	0	4	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	3	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	2	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	–1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
1	1	1	1	0	–2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
1	1	1	0	1	–3	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
1	1	1	0	0	–4	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
1	1	0	1	1	–5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
1	1	0	1	0	–6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	1	–7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	0	0	0	–8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	X	X	X	–8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



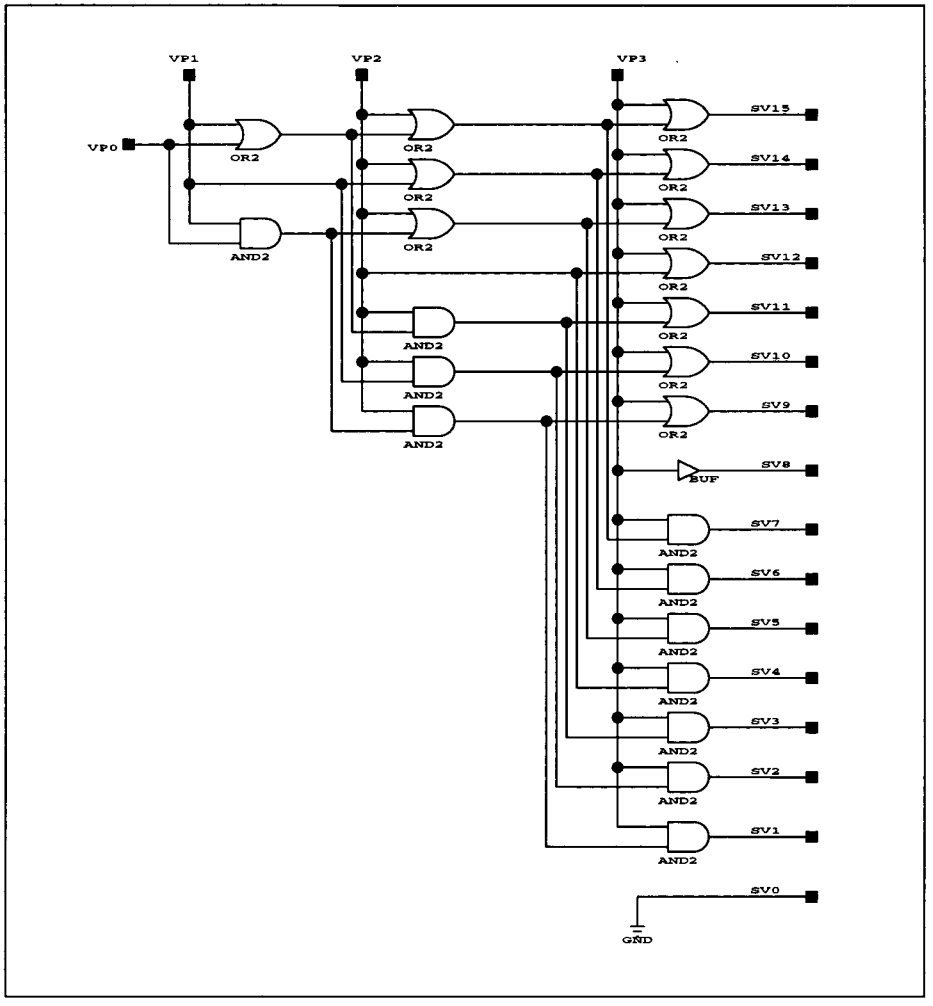
III. Top Level Schematic Printed From Viewlogic



IV. Logic for Limiter and Thermometer Coder (0^{th}-order ESL)



V. Logic for 1st-order ESL



VI. Logic for Thermometer Coder, 1st-order ESL

VII. Schematic of the Test Board